

Improving Aggregate User Utilities and Providing Fairness in Multi-rate Wireless LANs

by

Godfrey Tan

Bachelor of Science in Electrical Engineering and Computer Science,
University of California at Berkeley (1999)

and

Master of Science in Electrical Engineering and Computer Science,
Massachusetts Institute of Technology (2002)

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degrees of

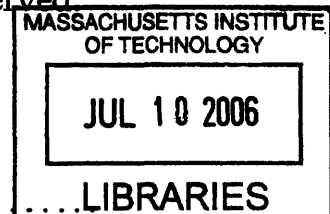
Doctor of Philosophy in Electrical Engineering and Computer Science,
Massachusetts Institute of Technology

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2006

© Massachusetts Institute of Technology 2006. All rights reserved.



Author
Department of Electrical Engineering and Computer Science
October 31, 2005

ARCHIVES

Certified by
John Guttag
Professor
Thesis Supervisor

Accepted by
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Improving Aggregate User Utilities and Providing Fairness in Multi-rate Wireless LANs

by
Godfrey Tan

Submitted to the Department of Electrical Engineering and Computer Science
on October 31, 2005, in partial fulfillment of the
requirements for the degrees of
Doctor of Philosophy in Electrical Engineering and Computer Science,
Massachusetts Institute of Technology

Abstract

A distributed medium access control (MAC) protocol is responsible for allocating the shared spectrum efficiently and fairly among competing devices using a wireless local area network. Unfortunately, existing MAC protocols, including 802.11's DCF, achieve neither efficiency nor fairness under many realistic conditions.

In this dissertation, we show that both bit and frame-based fairness, the most widely used notions, lead to drastically reduced aggregate throughput and increased average delay in typical environments, in which competing nodes transmit at different data transmission rates.

We demonstrate the advantages of time-based fairness, in which each competing node receives an equal share of the wireless channel occupancy time. Through analysis, experiments on a Linux test bed, and simulation, we demonstrate that time-based fairness can lead to significant improvements in aggregate throughput and average delay.

Through a game theoretic analysis and simulation, we also show that existing MAC protocols encourage non-cooperative nodes to employ globally inefficient transmission strategies that lead to low aggregate throughput. We show that providing long-term time share guarantees among competing nodes leads rational nodes to employ efficient transmission strategies at equilibriums.

We describe two novel solutions, TES (Time-fair Efficient and Scalable MAC protocol) and TBR (Time-based Regulator) that provide time-based fairness and long-term time share guarantees among competing nodes.

TBR is a backward-compatible centralized solution that runs at the AP, works in conjunction with DCF, and requires no modifications to clients nor to DCF. TBR is appropriate for existing access point based networks, but not effective when nearby non-cooperative nodes fall under different administrative domains. Our evaluation of TBR on an 802.11b/Linux test bed shows that TBR can improve aggregate TCP throughput by as much as 105% in rate diverse environments.

TES is a non-backward compatible distributed contention-based MAC protocol that is effective in any environment, including non-cooperative environments. Furthermore, the aggregate throughputs sustained with increased loads. Through extensive simulation experiments, we demonstrate that TES is significantly more effi-

cient(as much as 140% improvement in aggregate TCP throughput) and fairer than existing MAC protocols including DCF.

Thesis Supervisor: John Guttag

Title: Professor

Acknowledgments

A little over five years has passed since I began my journey as a Course-VI Ph.D. student at MIT. It has been challenging, rewarding and above all fun. I feel fortunate to have been a member of the Networks and Mobile Systems group. I am most grateful to John Guttag for supervising both my master's and doctoral theses. It has been an incredible opportunity to work with John, and I can sincerely say that I could not have asked for a better advisor. With John, I worked in a critical yet nurturing environment that allowed me to pursue my various interests in a supervised yet independent manner. As an excellent teacher and friend, John taught me many skills ranging from writing and presentation to thought-organizing and problem-solving. I will dearly miss our weekly stimulating meetings as well as our casual conversations on various political, social and investment related topics.

I am grateful to my thesis readers, Hari Balakrishnan and Dina Katabi. I worked closely with Hari on a few research projects and had many stimulating discussions that greatly improved my research skills. Hari offered sound advice on various technical issues and encouraged me in many ways. Dina significantly helped improve a hard-to-read chapter on my thesis. I benefited greatly from their friendship, willingness to help and precise feedback. I learned important presentation and problem-solving skills from Frans Kaashoek as well. I had an opportunity to work with him on one project, and learned tremendously from many interactions with him. Recommendations from Hari, John and Frans are the key in landing me an Intel Ph.D. Fellowship and getting me job interviews at various research laboratories and companies.

From Samuel Madden and Robert Morris, I learned many effective techniques for explaining and presenting complicated technical ideas in simple and intuitive ways. Dorothy Curtis and Michel Goraczko deserve all the credits for ensuring that I had all the resources that I needed to carry out my research.

My experience at MIT is enriched by my friends and colleagues with whom I interact daily. Allen Miu, a close friend and ideal collaborator, deserves a special token of appreciation and gratitude. Allen has been willing to entertain my various research ideas and many outrageous business ideas. Never discouraging and always engaging with patience, Allen has been a consistent source of sound feedback, and helps shape my raw scattered ideas into clearer focused ones. I hope to continue our strong friendship and perhaps one day start a business venture together.

I also learned many social and inter-personal skills from my friend and officemate, Magdalena Balazinska. Without Magda and Michel, interactions among NMS group members would not be as frequent or fun. I am going to miss taking coffee breaks

with Magda. My wife and I are thankful for various rides that Michel and Magda provided us during our stay in Boston area, and will miss their company.

Daniel Albadi, Kyle Jamieson and Asfandyar Qureshi have been excellent office-mates and friends. Our office is often active with conversations covering a diverse range of topics from NBA to social issues of South Asian nations. I value the friendship of Vladimir Bychkovskiy, Bret Hull, Eugene Shih and Stan Rost. The two offices that they reside are the ones that I frequently visit for no reasons but for company to chit-chat. I always looked forward to our many lunches together, which were typically convened at a long table in the lobby and noise-filled with personal anecdotes and insights on current social, economical and political issues.

I benefited from collaborations with Greg Harfst, Jaeyeon Jung, Russ Tedrake and Jessica Tseng in various class projects. I learned a great deal from *cool* research carried out by Dave Andersen, Nick Feamster, Bodhi Priyantha and Alex C. Snoeren. My experience at MIT is enriched by many fun conversations with Chuck Blake, Wenjun Hu, Srikanth Kandula, Sachin Katti and Ali Shoeb. I look forward to keeping up the friendship with all the wonderful people that I have come to know, and hope that I will have many chances to collaborate with them in future endeavors.

The older I grow, the more I become aware of each of the many positive influences that my parents have on me. Many agree that I got the drive to strive for high goals and intelligence from my mom. She leads by example and instills in me the value of working hard. My dad is a source of calmness, always putting life in a bigger perspective and never letting misfortunes bother him. Without his constant insistence that we learned English since we were relatively young, I could not imagine how I could have done well in my studies in America. I am also fortunate to have three great siblings who typically tolerate my irrational behaviors and let me win on countless occasions. I am especially grateful to my elder brother, Jeffrey, for always watching out for me. I appreciate continuing encouragement and support from my many uncles, aunts and cousins. It is such a great feeling to be a part of dozens of relatives who are constantly pondering ways to feast together. I am especially indebted to aunts Susie and Lucy for always being there for me, Uncle Lit and Aunt Rosemary for kindness and unwavering support, and Uncle Teddy's family for happily hosting Jeff and I for several months.

My wife, Lynn, has been the main source of my energy and optimism. With enduring love and companionship, she has supported me in every possible way. I may never know some of the many sacrifices that she made along the way. But, I know that my life would not have been the same without her.

Lastly, it was my late grandma, who was instrumental in ensuring that Jeff and I got the chance to leave Burma and obtain college education in America, four years earlier than when the rest of our family joined us. Each time I think about her, I am amazed at how she managed to raise eight loving children, and how well she was liked and loved by friends and families alike. She provided to my brother and I the comfort, love and kindness that we dearly needed, when we were thousands of miles away from our comfort zone. It is unfortunate that she never saw me in a graduation gown. For all that she has given to me, I devote this dissertation to *Ann Ma*, my loving grandmother.

Contents

1	Introduction	19
1.1	Goals: Use Spectrum Efficiently and Fairly	20
1.1.1	Efficiency	21
1.1.2	Fairness	21
1.2	The Problems	21
1.2.1	Poor Efficiency in the Presence of Rate Diversity	21
1.2.2	Unfairness In the Presence of Varying Channel Conditions . .	23
1.2.3	Poor Efficiency in the Face of Non-cooperative Competition .	23
1.2.4	Poor Efficiency in the Presence of Many Contenders	26
1.3	Causes of the Problems and Proposed Solutions	27
1.3.1	Frame-based Fairness is Inefficient	27
1.3.2	Solution #1: Time-based Fairness	29
1.3.3	Use-it-or-lose-it Policy Leads to Unfairness	29
1.3.4	Frame-based Fairness and Use-it-or-lose-it Policy Lead Nodes to Employ Inefficient Transmission Strategies	30
1.3.5	Solution #2: Long-term Time Share Guarantees	31
1.3.6	DCF's Distributed Channel Access Mechanism Does not Scale	31
1.3.7	Solution #3: A Scalable Distributed Channel Access Mechanism	32
1.4	Contributions and Dissertation Structure	33
1.4.1	Concepts and Analysis	33
1.4.2	Practical Solutions	33
1.4.3	Organization	35
2	Background and Related Work	37
2.1	Background	37
2.1.1	Communications in Unlicensed Frequencies	37
2.1.2	Rate Diversity is Prevalent	38
2.1.3	Evidence of Multiple Users During Congested Periods	39
2.1.4	Loss Avoidance at the Physical Layer	40
2.1.5	Loss Detection, Avoidance and Recovery at Link Layer	41
2.1.6	Fixed Assignment Techniques for Channel Access	44
2.1.7	Distributed Channel Access Protocols	45
2.1.8	802.11's Distributed Coordinating Function	46
2.1.9	Fair Queuing	48
2.2	Related Work	50

2.2.1	Collision Detection	50
2.2.2	Collision Resolution	51
2.2.3	Distributed Weighted Fair Scheduling	53
2.2.4	Distributed Time-based Fair Scheduling	54
2.2.5	Fair Share Guarantees and Compensation	55
2.2.6	Throughput, Delay and Equilibrium Analyses	56
2.2.7	Other Related Work	59
3	Analytical Model and Results	61
3.1	Fairness Notions for WLANs	61
3.1.1	Resource	62
3.1.2	Entities	63
3.1.3	Fairness Units	64
3.2	Throughput Analysis	64
3.2.1	Assumptions	65
3.2.2	Achieved Throughput	66
3.2.3	Representing Fairness Notions	68
3.2.4	Achieved MAC-layer Throughput Under Various Fairness Notions	69
3.3	Delay Analysis	70
3.3.1	Average Wait Time under Various Fairness Notions	70
3.3.2	Worst-case Relative Ratio of Wait Time in a Two-Entity Competition	72
3.4	Equilibrium Analysis	75
3.4.1	Game Model	76
3.4.2	Nash and Subgame Perfect Equilibriums	77
3.4.3	Error Model	78
3.4.4	Analysis of DCF	78
3.4.5	Analysis of EDCF	81
3.4.6	Providing Flexibility at the MAC Layer Improves Performance	84
3.4.7	Evidence of Inefficient Equilibria Through Simulations	87
3.4.8	Discussion	90
4	Comparison of Fairness Notions	93
4.1	Measuring Differences in Performance	93
4.2	Comparing Trade-offs between Performance and Relative Fairness	93
4.3	Throughput Comparison	95
4.4	Delay Comparison	98
4.5	Trace-driven Analysis of Wait Time	101
4.5.1	Workload Characteristics	102
4.5.2	Impact of Allocation Strategy on Wait Time	103
4.6	Impact of Greedy Channel Occupancy Time Allocations	105
4.7	Summary and Discussion	106

5	Distributed TES MAC Protocol	109
5.1	Definitions	110
5.2	Analysis on Achieving Optimal Utilization	111
5.2.1	Computing CW to Achieve a Desired Collision Rate	112
5.2.2	Relationships among $Tidle$, CW and $Ncont$	113
5.2.3	Maximizing Collision-free Channel Utilization	115
5.3	Overview of TES	116
5.4	Monitoring Transmission Events	118
5.5	Achieving High Collision-free Channel Utilization	118
5.6	Achieving Long-term Time-based Fairness	122
5.6.1	Configuring a Desired Fair Channel Time Share	124
5.6.2	Maintaining Fairness for Long-lived Active Links	125
5.7	Evaluation	127
5.7.1	Benefits of Time-based Fairness	128
5.7.2	Benefits of Per-link Fairness	129
5.7.3	Variations in Frame Size	130
5.7.4	Achieving Scalability	131
5.7.5	Convergence To Fairness	134
5.7.6	Random Channel Losses	137
5.7.7	Burst Losses	140
5.8	Summary	142
6	Centralized Time-based Regulator	143
6.1	TBR: Design and Implementation	143
6.1.1	Scheduling Frame Transmissions	145
6.1.2	Computing Channel Occupancy Time	147
6.1.3	Keeping Channel Utilization High	148
6.1.4	An 802.11-based Implementation	150
6.2	Evaluation	151
6.2.1	Limitations and Extensions	152
6.3	Summary	153
7	Conclusions	155
7.1	Summary	155
7.2	Potential Future Research Work	156
7.3	Conclusion	157
A	Supplements to Chapter 3	167
A.1	Supplements to Section 3.4	167
A.1.1	Analysis of DCF	167
A.1.2	Analysis of EDCF	169
B	Supplements to Chapter 5	171
B.1	Discussion on Analysis and Validation	171
B.1.1	Validation	172

List of Figures

1-1	Experimentally achieved TCP throughputs of two competing nodes when i) both send data at 11 Mbps, ii) both send at 1 Mbps, and iii) one sends at 11 Mbps and the other at 1 Mbps.	22
1-2	TCP throughput achieved at various data rates in a simulated environment. RTS/CTS is used to minimize collisions due to hidden terminals and incurs about 20% overhead when data bits are transmitted at 11 Mbps. The received power thresholds for various data rates are based on the Orinoco 802.11b Gold Card data sheet [24].	24
1-3	802.11-based WLANs spotted within a one-squared mile area near Seattle downtown (courtesy of [88]). The white, red and green boxes are configured for open access, encryption-enabled private access, or commercial access respectively.	26
1-4	Aggregate UDP throughput as a function of the number of transmitters in a simulated environment.	27
1-5	Experimentally achieved TCP throughputs and channel occupancy time of two competing nodes when i) both sending data at 11 Mbps, and ii) one sending at 11 Mbps and the other at 1 Mbps. These experiments were carried out using Cisco's Airo cards.	28
1-6	Fraction of collision events relative to the total number of transmission events.	32
2-1	Fractions of bytes transferred at various data rates during three 90-minute student workshop sessions (WS) at MIT and an experiment (EXP-1)	39
2-2	Fraction of throughput achieved by the heaviest user during busy intervals at an AP at a Dartmouth dormitory.	40
2-3	An example illustrating DCF's backoff counters of competing nodes A and B	47
3-1	Decision tree of a stage game under Flex-1. Node i may choose not to transmit in which case node j may transmit. If node j does transmit, node i then transmits in node j 's turn.	86
3-2	TCP throughput achieved when using various fixed data rates and RBAR, an auto-rate protocol. Regions (A) and (B) are where rational nodes under DCF may use inefficient strategies when competing against nodes with lower loss rates (smaller transmission distances). .	88

3-3	n_0 and n_1 transmit to m_0 and m_1 respectively.	88
3-4	TCP throughput achieved by n_1 and the aggregate achieved throughputs under two pairs of strategies as a function of the distance between n_0 and m_0 . $(R, 2)$ denotes that n_0 uses RBAR and n_1 transmits at a fixed data rate of 2 Mbps. $Tot(R, 2)$ plots the aggregate throughputs. However, the most efficient strategy for n_1 is to transmit at 5.5 Mbps, which is what RBAR running at n_1 would do. Thus, (R, R) denotes the most efficient strategy pair which may not be used at equilibriums.	89
4-1	The improvement in aggregate expected wait time of all sessions achieved under TF over BF ($AggrDiff(TF, BF)$) and the PF ratio of TF and BF.	100
4-2	CDF of number of nodes during congested periods.	101
4-3	CDFs of session size and inter-arrival time of actual trace and other traffic distributions using the same mean values obtained from the trace.	102
4-4	Distributions of wait time of 11 Mbps and 1 Mbps sessions. The data rate used by each session in this experiment is uniformly distributed among four 802.11b data rates.	103
4-5	Distributions of wait time of all sessions. The data rate used by each session in this experiment is uniformly distributed among four 802.11b data rates.	104
4-6	CDF of wait time of sessions as large as 1 MByte for two different distributions of 802.11b transmission speeds.	105
4-7	The improvement in aggregate expected wait time of all sessions achieved under TF over BF ($AggrDiff(TF, BF)$) and the PF ratio of three pairs of fairness notions.	106
5-1	An example illustrating two different types of transmission events: a successful transmission event and a collision event. There are 3 frame transmissions but only 2 transmission events.	110
5-2	Node B observes node A 's frame transmission every CW_A^{avg} idle timeslots. Shaded slots represent the slots in which node B senses the busy channel.	112
5-3	Pseudo-code of TES running at each node. Figure 5-4 describes remaining Procedures mentioned here.	117
5-4	Pseudo-code of backoff instance i	120
5-5	Pseudo-code of backoff instance i that provides long-term fair share guarantees. Figure 5-6 describes PROCEDURE UPDATETLAG.	123
5-6	Pseudo-code of backoff instance i that concerns with fairness.	126
5-7	Aggregate UDP and TCP throughputs achieved in either the uplink direction (11b-1a) or downlink direction (11b-1b) by four competing 802.11b clients, two sending at 11 Mbps and the other two at 1 Mbps.	128

5-8	Aggregate UDP and TCP throughputs achieved by four competing 802.11b clients, each exchanging data at 11 Mbps. In 11b-1c, there are 2 uplink and 2 downlink clients. In 11b-1d, there are 1 uplink client and 3 downlink clients. Although the aggregate throughput is roughly the same for both scenarios, fairness is affected (see Figure 5-9) . . .	129
5-9	Max/min ratio and time fairness index of two scenarios with UDP-only flows. In each scenario, TES achieves a very high degree of fairness, with both the max/min ratio and fairness index close to 1. DCF allocates less channel occupancy time to downstream links, leading to much higher degrees of unfairness among competing links.	130
5-10	Aggregate UDP and max/min time ratios when there are four competing 802.11b clients, with two using a payload size twice that of the other two.	131
5-11	Aggregate UDP Throughput as a Function of the Number of Contenders.	131
5-12	Measures related to efficiency and fairness when different numbers of backlogged nodes are sending UDP data packets to a common AP. . .	132
5-13	Evolutions of various measures over the first 19 second since 10 nodes simultaneously competed for channel access.	133
5-14	Evolutions of various measures over the first 19 second since 50 nodes simultaneously competed for channel access.	135
5-15	Evolutions of various measures over the first 19 second since 100 nodes simultaneously competed for channel access.	136
5-16	Evolutions of various measures over the first 19 second since 200 nodes simultaneously competed for channel access.	137
5-17	Jain's Fairness Index of Received UDP Packets.	138
5-18	Measures related to efficiency and fairness when nodes experience random losses because of channel errors, under three different scenarios. .	139
5-19	Measures related to efficiency and fairness when nodes experience losses in bursts because of channel errors, under two different scenarios. . .	141
6-1	Pseudo-code of most of TBR.	145
6-2	Pseudo-code of TBR that keeps track of the channel occupancy time allocated to each client.	147
6-3	Pseudo-code of the token rate adjustment event	149
6-4	TCP throughputs achieved in either uplink or downlink direction by two competing nodes using the same data rate. <i>Exp-DCF</i> and <i>Exp-TBR</i> denote the experiments that were run with the AP equipped without or with TBR respectively. $n_i(11)$ denotes the throughput achieved by node i transmitting at 11 Mbps.	150
6-5	TCP throughputs achieved in either uplink or downlink direction by two competing nodes using different data rates. <i>Exp-DCF</i> and <i>Exp-TBR</i> denote the experiments that were run with the AP equipped only with DCF without TBR and DCF with TBR respectively.	151

B-1	Ratios of the value obtained through simulation and the analytically derived value of $Pcol_{all}$, $Ntxpercol$, $Ftxev$ and $Tidle$, when 2 or 4 continuously backlogged nodes compete for channel access using the same CW and $Base = 1$	172
B-2	Absolute values of $Pcol_{all}$ and $Ntxpercol$ obtained through simulation when 4 continuously backlogged nodes compete for channel access using the same CW and $Base = 1$	173
B-3	Ratios of the value obtained through simulation and the analytically derived value of $Pcol_{all}$, $Ntxpercol$, $Ftxev$ and $Tidle$, when 2 or 4 continuously backlogged nodes compete for channel access using the same CW and $Base = 0$	173

List of Tables

1.1	Under DCF, the fractions of frame transmission among two nodes observing different channel conditions differ.	23
1.2	The aggregate TCP throughput (in simulation) is highest when n_0 and n_1 transmit at 11 and 5.5 Mbps respectively. However, at steady state, n_1 lowers its data rate to 2 Mbps to achieve higher throughput, while significantly degrading the aggregate throughput. n_0 cannot benefit by lowering its data rate and thus transmits at 11 Mbps at steady state.	25
1.3	When n_1 transmits at 2 Mbps, the channel occupancy time per successful TCP data frame transmission (7.14 ms) is about twice as large as when it transmits at 5.5 Mbps, showing that transmitting at a 5.5 Mbps is a more efficient strategy. However, with a reduction in frame error rate, n_1 will choose a less efficient strategy.	30
3.1	Comparison of the wait time of a session that completes no later than another session under F_i and that under F_j	74
3.2	Theoretically achievable UDP throughputs under all possible 802.11b data rates.	80
3.3	Frame success rates and corresponding practically achievable throughputs of node i and j at all possible transmission strategies.	80
3.4	Fractions of channel occupancy time and achieved throughputs of node i and j when each node is using either g_{11} or $g_{5.5}$. The unique NE strategies for i and j are $g_{5.5}$ and g_{11} respectively.	81
3.5	Frame success rates and corresponding practically achievable throughputs of node i and j at all possible transmission strategies under sEDCF with BFL.	83
3.6	Fractions of channel occupancy time and achieved throughputs of node i and j when each node is using either g_{11} or $g_{5.5}$. The unique NE strategies for i and j are $g_{5.5}$ and g_{11} respectively.	84
4.1	The theoretically achievable MAC-layer throughput (in Mbps) and the fair share of channel occupancy time of an entity (ϕ) running at each of the four possible 802.11b data rates (in Mbps).	96
4.2	Simulation results and analytically derived UDP throughputs of an experiment in which node 11 sends UDP data packets at 11 and node 1 at 1 Mbps.	97
4.3	<i>AggrDiff</i> and <i>PF</i> ratios for UDP experiments.	97

4.4	Simulation results and analytically derived TCP throughputs of an experiment in which node 11 exchanges TCP data with a receiver at 11 Mbps and node 1 exchanges TCP data with another receiver at 1 Mbps.	98
4.5	AggrDiff and PF ratios between TF and BF for TCP experiments. .	98
4.6	The practically achievable TCP throughput and the mean service rate, based on average session size of 294 KB, at each of the four possible 802.11b data rates. We assume that $\gamma^{prac,tcp}$ only depends on data rate, i.e., it remains the same under any fairness notion and any diverse mixes of data rate. We compute $\gamma^{prac,tcp}(d)$ by multiplying $\gamma^{theo}(d)$ with 0.65, i.e., the combined collision, idle time and TCP ack is assumed to be 35% of the run time.	99
5.1	Configurable TES parameters and their default settings.	127
6.1	Comparison of achieved TCP throughputs under Exp-DCF and Exp-TBR. Node $n2$ experienced the bottleneck bandwidth of 2.1 Mbps whereas node $n1$ could send as fast as it could (TCP permitted). Both nodes transmitted at 11 Mbps.	152

Chapter 1

Introduction

The 802.11 family is an increasingly popular wireless local area networking (WLAN) standard. In a typical deployment called the *infrastructure mode*, a mobile node or a client station equipped with an 802.11 interface communicates over the air using unlicensed frequency bands to an access point (AP) that is connected to a wired backbone.

Today, 802.11-based WLANs are deployed in many offices, buildings and homes. Common usages are downloading and uploading of web pages, email, files and interactive VOIP (voice over IP) data. When there are several users connected to a WLAN (e.g. at a conference or a hotspot), many often experience noticeably large network delay and/or low throughput. This diminished user utility is not usually caused by lack of achievable channel capacity. More often, the problem is that the 802.11 medium access control (MAC) protocol, DCF (for Distributed Coordination Function), which allocates the channel capacity among competing client nodes, is i) inefficient, leading to low aggregate throughput and high network delay and ii) unfair, leaving some clients with very small shares of network capacity and others very large shares. This problem will become worse as mobile users increasingly use throughput-intensive and delay-sensitive applications such as real-time video streaming and VOIP applications.

The problem of fair and efficient resource allocation in networks is not new. However, three important factors that are relevant to today's WLANs change the problem studied extensively in the context of wireline networks. First, today's WLANs are *rate diverse* in that nodes transfer data at a number of different transmission rates or speeds. Second, multiple nearby nodes may compete for channel access in a *rational* but *non-cooperative* manner. That is each competing node will attempt to maximize its utility regardless of the impact on other nodes. Third, the number of nodes that compete for channel access is often more than a few nodes, and increasing.

In the presence of these factors, we investigate through analysis and experiments, how various capacity allocation policies, including DCF's policy, affect the achieved throughput and observed delay of each client.

We identify the main reasons for inefficient and unfair allocations of the shared channel capacity under DCF and other existing MAC protocols. We then present new MAC-layer solutions that can significantly improve aggregate user utilities of WLANs

over existing practice.

We present our results in the context of 802.11-based networks running in the infrastructure mode. In the *infrastructure* mode, an AP acts as a *bridge* or a relay between its clients and the wired infrastructure. Each wireless client in an AP-based WLAN forwards and receives frames to and from its associated AP. In a typical configuration, the AP is attached to a wired Ethernet LAN and its associated clients appear as Ethernet hosts to other hosts on the LAN. Today, the infrastructure mode is the preferred mode for indoor communications. The alternative operating mode of 802.11 is the *ad hoc* mode, in which nodes can communicate in a peer-to-peer fashion without requiring the presence of APs. Today, the *ad hoc* mode is mainly used for connecting nodes in distant locations in community wireless mesh networks and experimental outdoor testbeds [4].

Although our work focuses on AP-based 802.11 WLANs, many of our findings and solutions are applicable to many types of WLAN technologies that use carrier-sensing distributed channel access protocols, such as sensor networks [28, 38], HomeRF-based networks [43] and HiperLAN/1-based networks [41].

The next section describes the desired goals in allocating the shared channel resource. Sections 1.2 and 1.3 detail how and why existing solutions are not adequate in meeting the desired goals in today’s environments respectively. Finally, Section 1.4 describes our contributions and a road-map to the rest of this dissertation.

1.1 Goals: Use Spectrum Efficiently and Fairly

A MAC protocol must allocate channel capacity among competing entities efficiently and fairly. Efficiency is important because the wireless spectrum available for wireless LANs is limited. Achieving fairness among competing entities has been a widely accepted “social goal.”

For a given frequency bandwidth (width of frequency band) and a signal to noise ratio, the theoretically achievable channel capacity is also limited [89]. Thus, the shared wireless spectrum must be used efficiently. With each generation of transmission technology, the achievable transmission speed improves significantly, inching closer to hitting the theoretical limit. For example, the maximum symbol transmission rate under 802.11b is 11 Mbps whereas that under 802.11g and 802.11a is 54 Mbps. However, due to significant physical layer and MAC layer overheads, under DCF, the aggregate UDP throughput of an 802.11b channel is only about 6 Mbps, and that of an 802.11a or 802.11g channel is only about 30 Mbps. Taking advantage of improvements in transmission speeds, future wireless applications including real-time mobile gaming and real-time mobile video conferencing applications will be more throughput-intensive. For example, an HDTV-compliant streaming application can demand up to 240 Mbps, assuming a minimum delivery rate of 30 frames per second, with each frame taking 1 MByte. Therefore, an efficient use of the scarce channel capacity is becoming more important.

Furthermore, when multiple devices share a common channel, the allocation of the shared channel resource among competing devices must be fair. A MAC protocol

must ensure that multiple competing devices share a common channel in a both fair and efficient manner.

1.1.1 Efficiency

A straight forward measure of a WLAN's performance is the aggregate throughput achieved by competing nodes. The achieved throughput of a node can be much smaller than its transmission speed for a variety of reasons including overheads incurred by both the MAC and physical layers and losses due to collisions and channel errors.

Another important measure is the *wait time* or *response time* of a user's task, which is the amount of time required to complete a task from the time the task is started. The average wait time is greatly impacted by the way in which channel occupancy time is allocated by the MAC protocol during congested periods. Busy periods in wireless networks can last from several milliseconds to several dozens of minutes [6, 60, 96], and thus improving user wait time is highly desirable. A MAC protocol must strive for both high aggregate throughput and low wait time.

1.1.2 Fairness

Abstractly, there are two major dimensions in defining a fairness notion for allocating the shared resource: *entities* and *fairness units*. Entities share the resource in the form of fairness units. A fairness notion dictates i) what the entities are (e.g., nodes or links), ii) what the fairness units are (e.g., frames or time), and iii) what the fair share of each entity is. We show in this dissertation that the choice of fairness notion can significantly impact the efficiency.

There exist well known measures to quantify unfairness [46, 59], which we will use to evaluate the effectiveness of a MAC protocol in providing fairness. A MAC protocol must strive, for each competing entity, to minimize variation in the actual share of channel resource allocated to it from its desired fair share.

1.2 The Problems

Today's WLANs operate in environments in which i) nodes transmit using a diverse set of data rates, ii) a large number of nodes compete for channel access, iii) nodes experience varying channel conditions and carry differing loads, and iv) nodes compete in a rational and non-cooperative manner. In this section, we demonstrate through experiments and simulations that 802.11's DCF is neither efficient nor fair under the impact each of these characteristics.

1.2.1 Poor Efficiency in the Presence of Rate Diversity

Nodes connected to 802.11 WLANs transfer data at a number of different rates or speeds for two major reasons: i) a sender can transmit at a lower data rate (using a more resilient modulation scheme) to reduce the channel bit error rate (BER) and

ii) different members of the 802.11 family co-exist in the same frequency band; for example, an 802.11b node with the maximum speed of 11 Mbps may compete against an 802.11g node with the maximum speed of 54 Mbps. Such environments are called *rate diverse*. We show empirical evidence of the prevalence of rate diversity in 802.11b WLANs in Section 2.1.2. In this subsection, we show that the aggregate throughput under DCF degrades significantly in the presence of rate diversity.

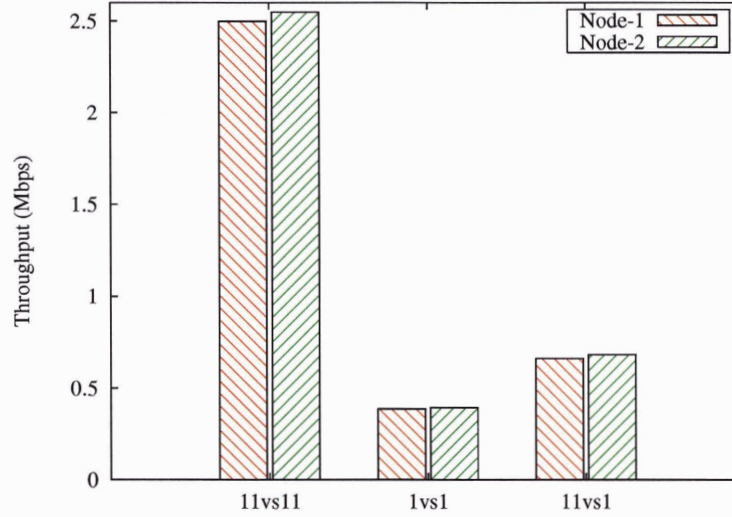


Figure 1-1: Experimentally achieved TCP throughputs of two competing nodes when i) both send data at 11 Mbps, ii) both send at 1 Mbps, and iii) one sends at 11 Mbps and the other at 1 Mbps.

The maximum achievable throughput of a channel varies with the transmission rates used by competing nodes. For instance, the aggregate TCP throughput of a pair of nodes transmitting at 11 Mbps is about 5.08 Mbps and is 0.78 Mbps for a pair transmitting at 1 Mbps. In each case, the aggregate TCP throughput is significantly lower than the transmission rate. This is because of large MAC layer and physical layer overheads and relatively smaller network layer and transport layer overheads. Also, the ratio of the aggregate TCP throughput to the transmission rate is much lower when nodes are transmitting at 11 Mbps than when they are transmitting at 1 Mbps. This is because a significant fraction of the overheads incurred by both the 802.11 MAC and physical layers is constant for each frame transmission, irrespective of data transmission rates. For instance, for each data frame transmission, the channel occupancy time required to transmit the physical layer preamble is constant since it is always transmitted at 2 or 1 Mbps (for robustness), irrespective of how long it takes to transfer data bits. Therefore, the overhead as a fraction of the total channel occupancy time required to transmit a frame is higher when the frame is transmitted at a higher data rate than when it is transmitted at a lower data rate.

When multiple nodes are simultaneously exchanging data using different data

	Fraction of Frame Transmissions	Loss Rate (%)	UDP Throughput
$n0$	0.561	4.39	4.029
$n1$	0.439	10.30	2.956

Table 1.1: Under DCF, the fractions of frame transmission among two nodes observing different channel conditions differ.

rates during congested periods, the total network throughput is quite different from what one might expect. Figure 1-1 illustrates how the aggregate throughput can degrade dramatically when two competing nodes transmit at different data rates, one at 11 Mbps and the other at 1 Mbps, when uploading files using TCP. One might expect the total throughput of an 11 Mbps and a 1 Mbps channel to be somewhere around 2.93 Mbps, the average of the total throughputs achieved by a pair of 11 Mbps channels (5.08 Mbps) and a pair of 1 Mbps channels (0.78 Mbps). However, for reasons discussed later, it is only 1.34 Mbps, less than half of what one might expect. The situation is likely to become worse as emerging 802.11g networks, with the maximum data rate of 54 Mbps, are deployed alongside relatively slower 802.11b networks. If no changes are made to MAC protocols, 802.11g users will see much lower performance improvement than expected.

1.2.2 Unfairness In the Presence of Varying Channel Conditions

Channel conditions can vary widely among transmitting and receiving pairs of nodes [64, 71, 83]. This (often) uncorrelated variations in observed channel conditions among different transmitting and receiving node pairs are related to differences in locations and mobility characteristics of nodes and the transmission techniques used (e.g., coding, frame size, etc.). Varying channel conditions among nodes lead to different observed frame loss rates among nodes. In this subsection, we show through simulations that in the presence of channel errors, DCF leads to unfairness.

Table 1.1 shows the results of a simulation when two nodes $n0$ and $n1$ send UDP data to a common AP. $n0$ and $n1$ are about 3 m and 18 m away from the AP respectively. $n0$, which suffered a frame loss rate of 4.39% transmitted 27% more frames and achieved 36% higher UDP throughput than $n1$, which was observing a frame loss rate of 10.3%. Such unfairness is undesirable in its own right and can sometimes also lead to degradation in aggregate throughput, as shown in the next two subsections.

1.2.3 Poor Efficiency in the Face of Non-cooperative Competition

When WLANs are deployed in non-cooperative environments, each competing rational node will attempt to maximize its utility (throughput or delay) regardless of

the impact on other nodes' throughput and delay. For instance, many WLAN users competing for channel access at a hot-spot such as a coffee shop or at neighboring apartments will be most interested in achieving the highest throughputs for themselves. For now, we define a *rational* node as one that will attempt to maximize its throughput by maximizing the product of its share of channel occupancy time and its achievable throughput per unit of occupancy. We will give a more formal definition in a later chapter. An 802.11 node can, for each frame transmission, set the frame size and the data transmission rate to attempt to maximize its utility. As we show in this section, existing MAC protocols can lead rational, non-cooperative nodes to inefficient equilibria, in which nodes intentionally use inefficient data rates or frame sizes.

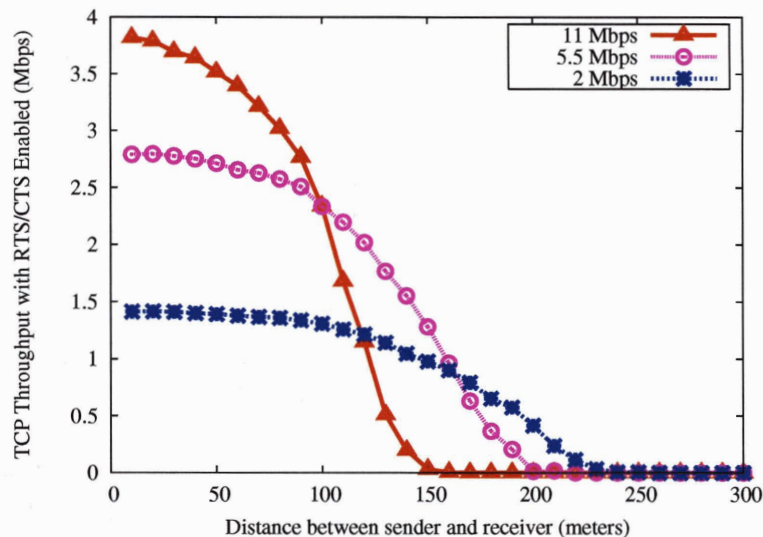


Figure 1-2: TCP throughput achieved at various data rates in a simulated environment. RTS/CTS is used to minimize collisions due to hidden terminals and incurs about 20% overhead when data bits are transmitted at 11 Mbps. The received power thresholds for various data rates are based on the Orinoco 802.11b Gold Card data sheet [24].

To illustrate the phenomenon, in which DCF leads rational nodes to use inefficient transmission strategies, we first explain how a rational node should choose its transmission based on perceived channel conditions. Figure 1-2 shows the achieved TCP throughput of a sender as a function of the distance between it and a receiver in a simulated environment using the *ns-2* network simulator [76]. RF propagation is modeled using a two-ray ground large-scale radio propagation model and a Rayleigh fast-fading model [80]. The latter models the fading phenomenon on short time-scales, which arises because of moving transmitters, receivers, and objects along transmission paths. In our simulation, propagation (and hence loss rate) is modeled only as a function of the distance between the sender and the receiver. Of course, in practice,

Data Rates (Mbps)	(11,5.5)	(11, 2)
$n0$'s throughput	3.219	2.243
$n1$'s throughput	0.491	0.579
Total (Mbps)	3.710	2.822

Table 1.2: The aggregate TCP throughput (in simulation) is highest when $n0$ and $n1$ transmit at 11 and 5.5 Mbps respectively. However, at steady state, $n1$ lowers its data rate to 2 Mbps to achieve higher throughput, while significantly degrading the aggregate throughput. $n0$ cannot benefit by lowering its data rate and thus transmits at 11 Mbps at steady state.

loss rates can be high even at short distances because of interfering objects such as thick walls etc.

For each pair of data rates, there exists a *cross-over distance* at which using a lower data rate yields higher throughput because the reduction in frame loss rate at the lower data rate is high enough to compensate for the slower transmission speed. For instance, as shown in Figure 1-2, at distances greater than 100 m, transmitting at 5.5 Mbps yields higher achievable throughput than transmitting at 11 Mbps. There are effective mechanisms for selecting an appropriate data rate for each frame transmission based on the received signal strengths [42] or the perceived loss rate [51].

Each competing rational node should choose the data rate that yields the highest achieved throughput under existing channel conditions. For example, in Figure 1-2, the cross-over distance defines the optimal transmission rate for senders. Unfortunately, this will not happen under DCF. Is the data rate that yields the highest achievable throughput also the data rate that leads to the highest achieved throughput? Not under DCF.

Consider a scenario in which each node uses maximum-sized data frames. Table 1.2 shows the achieved TCP throughputs of two sending nodes $n0$ and $n1$, each of which sends data to a different receiver. The distance between $n0$ and its receiver is 10 m whereas the distance between $n1$ and its receiver is 145 m. All nodes are within radio range of each other. As shown in Figure 1-2, in the absence of contention, the data rates that yield the highest achievable throughputs for $n0$ and $n1$ are 11 and 5.5 Mbps respectively. However, in the presence of competition, rational node $n1$ would lower its data rate to 2 Mbps to increase its achieved throughput by 18%. Unfortunately, this comes at the expense of reducing the aggregate throughput by 24%. $n0$ would not benefit by reducing its data rate, so it transmits at 11 Mbps at steady state.

In Section 3.4, we will revisit this issue in detail and show in a game theoretic setting that DCF leads rational nodes to use inefficient transmission strategies at equilibriums.

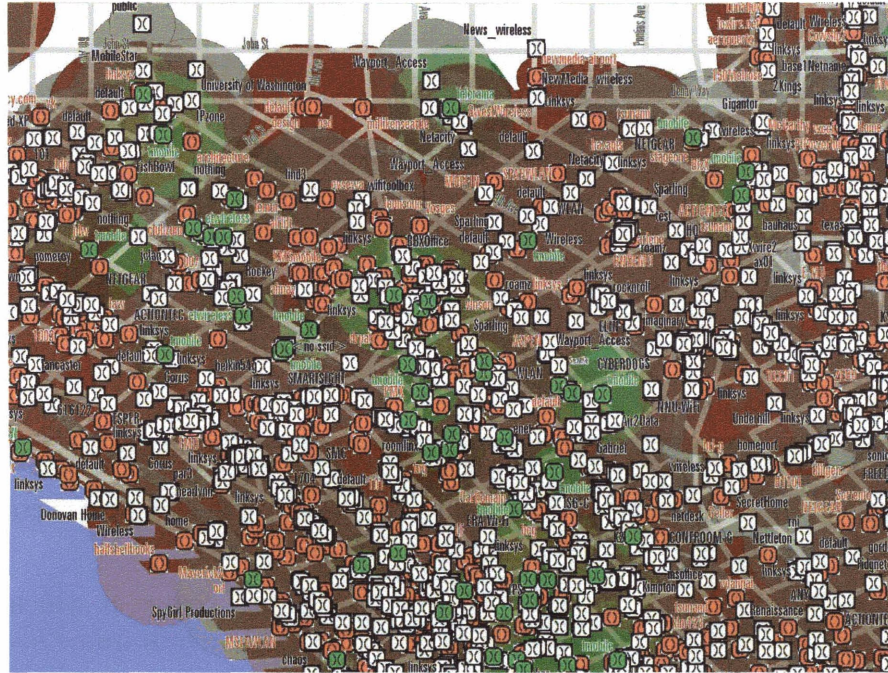


Figure 1-3: 802.11-based WLANs spotted within a one-squared mile area near Seattle downtown (courtesy of [88]). The white, red and green boxes are configured for open access, encryption-enabled private access, or commercial access respectively.

1.2.4 Poor Efficiency in the Presence of Many Contenders

In the past few years, 802.11-based WLANs have been deployed in both commercial and residential buildings at an astonishing pace. Figure 1-3 shows the map of existing 802.11-based WLANs in the Seattle (Washington) downtown area. Like Seattle, many cities in the U.S. and countries around the world have dense deployments of 802.11-based WLANs.

Ease of deployment and continued integration of 802.11 wireless interfaces into everyday devices have been leading to increased competition for channel access among nearby devices, which often fall under different administrative domains. With the limited nature of spectrum and the continued growths of mobile applications and 802.11-based devices, we expect the competition among non-cooperative devices to increase in the future.

The aggregate throughput under DCF significantly decreases with the number of contending nodes. Figure 1-4 shows how the aggregate UDP throughput decreases with the number of transmitters, each of which is continuously backlogged. The aggregate UDP throughput is maximum when there are only 2 contenders. As shown in the figure, the aggregate UDP throughput is reduced by about 8.3% when there are 10 contenders and by 26%, when there are 50 contenders. In practice, the number of contenders can be anywhere from a few nodes (e.g., a WLAN in a suburb home

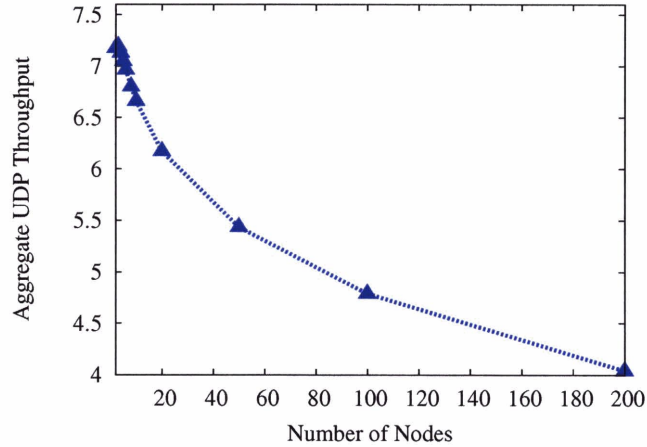


Figure 1-4: Aggregate UDP throughput as a function of the number of transmitters in a simulated environment.

or basement) to tens of dozens of nodes (e.g., WLANs in auditoriums, air planes or congested hotspots). As the number of 802.11 devices grow, the utility of 802.11-based WLANs will be significantly hampered because of the throughput degradation in the presence of a high number of contenders.

1.3 Causes of the Problems and Proposed Solutions

In this section, we explain the root cause of each of the problems mentioned in the previous section.

1.3.1 Frame-based Fairness is Inefficient

In Section 1.2.1, we showed that under DCF, when a node competes against another node transmitting at a lower data rate, its achieved throughput is much lower than what it would achieve when it competed against another node transmitting at the same speed. As a result, the aggregate throughput is much lower than what one would expect.

The root cause of this behavior is the fairness notion implied by DCF. DCF is designed to give approximately equal *transmission opportunities* (TXOPs) to each competing node. That is, to say each node will have approximately the same number of opportunities to send a data frame, irrespective of the amount of time required to transmit a frame. We call this *frame-based fairness*, under which allocation units are frames, entities are nodes, and nodes have equal priorities. When same-sized frames are used and channel conditions are similar, each competing node, regardless of its data rate, achieves roughly the same throughput under frame-based fairness.

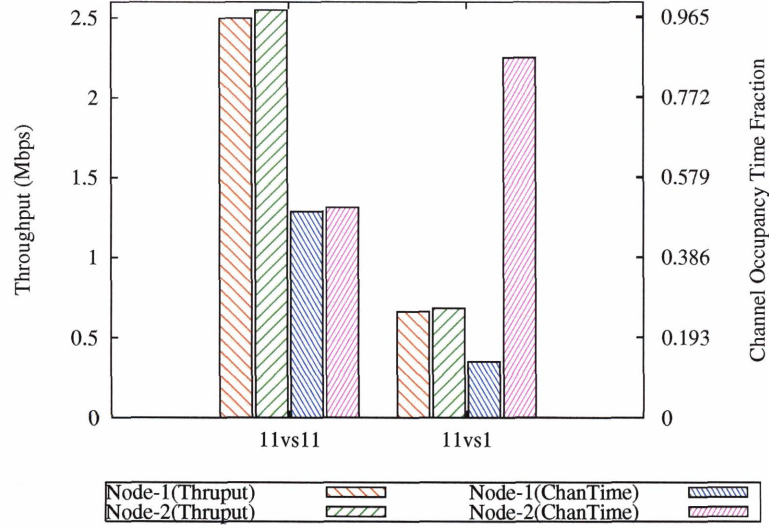


Figure 1-5: Experimentally achieved TCP throughputs and channel occupancy time of two competing nodes when i) both sending data at 11 Mbps, and ii) one sending at 11 Mbps and the other at 1 Mbps. These experiments were carried out using Cisco's Airo cards.

As shown in the two leftmost bars in Figure 1-5, when two nodes compete for channel access using 11 Mbps (the 11vs11 case), both achieve equal throughput. This is because both nodes achieve equal fractions of channel occupancy time as indicated by the remaining two bars in the 11vs11 case.

However, the channel occupancy time allocation is different when one node transmits at 11 and the other at 1 Mbps. Since the node transmitting at 1 Mbps will take several times longer to transmit a frame than the node transmitting at 11 Mbps, the channel is being used most of the time by the slower node. In Figure 1-5 (see the 11vs1 case), the fraction of the channel occupancy time used by the slower node is 6.4 times as much as that used by the faster node. Hence, the total throughput is reduced to a level much closer to what one gets when both competing nodes are slow. The faster node pays a penalty for competing against a slow node.

DCF mainly affects the channel capacity allocation in the uplink direction. The frame scheduling mechanism at the AP dictates the channel capacity allocation to clients in the downlink direction. When there are multiple backlogged frames destined to more than one client, the scheduling scheme must decide the order of transmission. The fairness notion used by most scheduling schemes in the literature [25, 37, 90] impacts the channel capacity allocation on the downlink direction in a similarly undesirable way as in the uplink direction as explained earlier.

1.3.2 Solution #1: Time-based Fairness

In our view, the fundamental shared resource for a given wireless channel is *channel occupancy time*, the time available to transfer data, and not bits or frame transmission opportunities.

Our analysis shows that *time-based fairness* is more efficient in terms of throughput and delay than many other existing fairness notions including frame-based fairness [94]. Under time-based fairness, allocation units are simply channel occupancy time units and thus entities with equal priorities will achieve equal shares of channel occupancy time.

Time-based fairness, unlike many other fairness notions, achieves the following desirable property:

The long-term throughput of an entity (node) competing against any number of entities (nodes) running at different speeds is equal to the throughput that the node would achieve in an existing single-rate 802.11 WLAN in which all competing entities (nodes) were running at its rate.

We call this the *independence property*, because under this property, the utility of an entity, in terms of achieved throughput and delay, is independent of the utilities of any other entities.

As shown earlier, under DCF's frame-based fairness, the aggregate TCP throughput is 1.34 Mbps when two nodes compete for channel access with one transmitting at 11 Mbps and the other at 1 Mbps. In contrast, under time-based fairness, the faster node competing against the slower node will achieve the same share of channel occupancy time as it would achieve when it competed against another node transmitting at the same speed of 11 Mbps. Therefore, its achieved throughput remains at about 2.5 Mbps, irrespective of the data rate used by its competitor, assuming that the MAC and transport layer overhead remains the same under both fairness notions. Similarly, the achieved TCP throughput of the slower node is about 0.39 Mbps. Thus, the aggregate TCP throughput (2.89 Mbps) under time-based fairness is 116% more than that (1.34 Mbps) under frame-based fairness.

Fairness is, of course, a subjective notion. We do not claim that one notion is "fairer" than the other. However, in the presence of rate diversity during congested periods, time-based fairness does improve the overall network performance when compared to many other traditionally accepted fairness notions.

1.3.3 Use-it-or-lose-it Policy Leads to Unfairness

In Section 1.2.1, we showed that under DCF, nodes observing different loss rates are allocated different number of frame transmission opportunities. The immediate cause of this unfairness is DCF's *exponential backoff* mechanism, which exponentially reduces the transmission probability of a node with each successive loss encountered. Therefore, over time, a node observing a higher loss rate will be given a lesser number of transmission opportunities than a node observing a lower loss rate. This backoff mechanism is critical in avoiding collisions in a distributed fashion, and is beneficial in

Data Rates (Mbps)	(11,5.5)	(11,2)
Channel occupancy time per TCP data frame (ms)	(1.75, 3.54)	(1.75, 7.14)
Error rate of TCP data frames (%)	(0.2, 15.9)	(0.2, 2.1)

Table 1.3: When $n1$ transmits at 2 Mbps, the channel occupancy time per successful TCP data frame transmission (7.14 ms) is about twice as large as when it transmits at 5.5 Mbps, showing that transmitting at a 5.5 Mbps is a more efficient strategy. However, with a reduction in frame error rate, $n1$ will choose a less efficient strategy.

the presence of channel errors that are highly correlated on short time-scales. When a node suffers a frame loss because of bad channel conditions, it is highly likely that its subsequent frame transmission will also fail. In such situations, a node experiencing a “bad channel state” should delay its transmissions for some milliseconds and let other nodes that are experiencing “good channel states” transmit, thereby improving the overall efficiency. A node running DCF will be forced to do so since the exponential backoff scheme is the standard. However, those potential transmission opportunities are forever lost since the node’s future transmission probability is not increased to compensate for the lost opportunities. We call this policy *use-it-or-lose-it*. The use-it-or-lose it policy when used in combination with an exponential backoff mechanism or alike leads to unfairness. The *use-it-or-lose-it* policy can also lead to degraded throughput in the context of non-cooperative competition, as explained in the next subsection. We propose a solution to both problems in Section 1.3.5.

1.3.4 Frame-based Fairness and Use-it-or-lose-it Policy Lead Nodes to Employ Inefficient Transmission Strategies

In non-cooperative environments, frame-based fairness and/or use-it-or-lose-it policy lead rational nodes to employ inefficient transmission strategies. Under DCF, the share of channel occupancy time obtained by a node depends on the data rates used by it and its competitors. In the example described in Section 1.2.3, by intentionally transmitting at a lower data rate, node $n1$ achieved a higher channel occupancy time share than it would by transmitting at a higher and more efficient data rate. This effect combined with the slight reduction in the node’s frame loss rate (due to using a more robust transmission speed) may lead to higher achieved throughput for that node. However, this is done at the cost of overall efficiency. The aggregate throughput lost by other nodes will exceed the throughput gained by the node using the inefficient transmission rate.

Table 1.3 shows the channel occupancy time per TCP data frame and the frame error rate, i.e., the frame loss rate due to channel errors (excluding collisions), of node $n0$ and node $n1$ for the scenario described in Section 1.2.3. For node $n1$, transmitting at 5.5 Mbps is more efficient than transmitting at 2 Mbps. That is, if $n1$ alone is occupying the channel, the number of successful frame transmissions at 5.5 Mbps will be more than that achieved at 2 Mbps. However, in the presence of competition $n1$

is allocated the same number of transmission opportunities irrespective of the data rate used and thus opts to choose a data rate that can improve its frame loss rate. However, $n1$'s improvement is achieved at the expense of overall efficiency.

Although the problem illustrated here is because of unequal channel occupancy time allocations as dictated by frame-based fairness, this problem also applies to other fairness notions like time-based fairness if fairness units are allocated in a use-it-or-lose-it manner. As explained in the earlier subsection, in the presence of time-correlated channel errors, a rational node should attempt to use an efficient transmission strategy by dictating not only what data rate and frame size to use but also *when to transmit*. However, under the use-it-or-lose-it policy, a node that gives up its potential transmission opportunities will not be given replacement TXOPs. Thus, a rational node will not voluntarily give up its transmission opportunities even when encountering avoidable losses that occur in bursts. Instead, it will try to improve its achieved throughput by transmitting (probably at a lower data rate) leading to a suboptimal use of the shared channel.

1.3.5 Solution #2: Long-term Time Share Guarantees

In general, a MAC protocol could lead rational nodes to inefficient equilibria if it leaves opportunities for a rational node to use a less efficient transmission strategy to get a higher share of allocation units than it would normally get when using a more efficient transmission strategy. A MAC protocol should eliminate such opportunities, thereby allowing rational nodes to improve their utilities only by using more efficient transmission strategies.

A MAC protocol can eliminate such opportunities by i) providing a time-based fairness notion and ii) guaranteeing the allocation of long-term fair shares of channel occupancy time to competing nodes. In Chapter 3, using a game theoretic analysis and through simulation, we show that when a MAC protocol meets these two conditions, rational non-cooperative nodes will choose efficient transmission strategies, irrespective of the nature of competition for channel access and channel conditions.

1.3.6 DCF's Distributed Channel Access Mechanism Does not Scale

In centralized channel access schemes like TDMA (for time-division multiple access), collision is avoided because the central entity (e.g., the base station) can schedule channel access among clients in a pre-determined fashion so that no two nearby clients will transmit at the same time. TDMA is widely used in cellular telephony networks. However, such a scheme requires frequent coordination among competing clients. This is possible in cellular networks because in each cellular network, no other nodes other than the clients of that network have the legal right to use the wireless channel, which was allocated to the network operator through a public spectrum auction process.

In contrast, anyone owning an 802.11-based device can transmit and receive data in the unlicensed band and explicit coordination among nearby devices, which may

potentially fall in different administrative domains, may not be possible for many reasons including privacy and security. Therefore, distributed channel access protocols are preferred in WLANs. DCF is such a mechanism.

As shown in Section 1.2.4, under DCF, the aggregate throughput decreases with the number of contenders. This is because the collision rate increases with the number of contenders as shown in Figure 1-6.

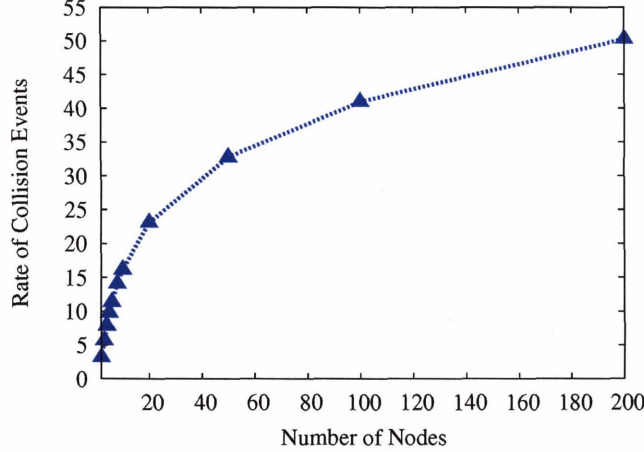


Figure 1-6: Fraction of collision events relative to the total number of transmission events.

This example illustrates that DCF's distributed channel access mechanism does not scale with the number of contenders. Many MAC protocols like DCF use randomized channel access schemes to reduce collisions when there is more than one node competing for channel access [2, 9, 29, 34, 48, 64, 82, 102]. In general, under these protocols, each node attempts to transmit with a certain probability. If the number of contenders increases significantly but this probability of transmission does not decrease in a similar fashion, the overall collision rate will increase.

As shown in Figure 1-6, it has been observed that DCF's distributed channel access mechanism fails to appropriately increase each node's transmission probability accordingly when the number of nodes is more than a few, leading to increased collisions and reduced aggregate throughput [15, 97, 98].

1.3.7 Solution #3: A Scalable Distributed Channel Access Mechanism

In Chapter 5, we describe a novel distributed channel access mechanism that appropriately adjusts the probability of transmission at each node even when the number of contenders is large. Our protocol can ensure that the collision rate is lower than a pre-configured target collisions rate with little or no overhead, irrespective of the number of contenders. Our evaluation shows that compared to DCF, our protocol can

achieve about 12%, 25% and 40% gains in aggregate throughputs when the number of contenders is 20, 50 and 100 respectively.

1.4 Contributions and Dissertation Structure

The main contributions presented in this dissertation are: i) extensive analyses of the impact of various fairness notions on performance and of the trade-offs between fairness and performance, ii) effective and practical link-layer solutions that use the limited spectrum in a fair, efficient and scalable fashion, and iii) an evaluation of the solutions using a Linux testbed and the *ns-2* simulator and a comparison against many existing solutions.

1.4.1 Concepts and Analysis

We propose that each competing entity be allocated an equal amount of channel occupancy time, not frames or bits, since channel occupancy time is the fundamental shared resource. Based on our analytical results [94] employing time-based fairness in 802.11b networks can improve aggregate UDP throughput by as much as 180% over frame-based fairness. Average user wait time also improves significantly. We also develop new measures to quantify the trade-offs between performance gains and “degree of unfairness” between two different notions of fairness. We conduct a trace-driven analysis to compare the average user wait time achieved under different fairness notions and evaluate the trade-offs of one fairness notion over another using our measures. Our results clearly show that compared to DCF’s frame-based fairness, time-based fairness improves the network efficiency significantly.

We also argue that each competing entity should have a long-term guarantee on its fair share of channel occupancy time. That is the MAC protocol should provide equal amounts of channel occupancy time among competing entities even when they are experiencing varying channel conditions and employing schemes to avoid time-correlated losses; such a scheme usually reduces the transmission probability of a node upon encountering a frame loss.

We develop a game theoretic model and show that i) a MAC protocol such as DCF can lead rational nodes to inefficient equilibria [93, 95] and ii) a MAC protocol that provides long-term time share guarantees can lead rational nodes to use efficient transmission strategies at equilibria.

1.4.2 Practical Solutions

We develop two different solutions, TES and TBR, that achieve a time-based fairness notion and provide channel occupancy time share guarantees, for two different target environments.

Time-fair Efficient and Scalable MAC Protocol

TES is a distributed MAC protocol that is effective in any environment, including non-cooperative environments where explicit coordination among nearby devices is not possible. TES achieves time-based fairness and provides long-term time-share guarantees among competing entities. Furthermore, TES is scalable in that the aggregate throughput is sustained with increased load as a result of achieving a pre-configurable collision rate, irrespective of the number of contenders or traffic mixes. TES is not a backward-compatible solution, i.e., TES will not directly benefit existing devices that run other MAC protocols such as DCF, since TES is intended as an alternative MAC protocol.

We implemented TES in the *ns-2* simulator [76]. Extensive simulations show that TES achieves the desired goals and significantly outperforms DCF. For example, when multiple 802.11b nodes compete for channel access using different data rates, TES improves aggregate throughput over DCF by as much as 170%. Furthermore, when the number of contenders is about 50, TES improves aggregate throughput over DCF by as much as 28%.

In addition to solving the main problems described in this chapter, TES allows for many link-layer optimization schemes that further improve the overall network efficiency by decoupling MAC-layer collision/congestion control from mechanisms achieving link-layer reliability. We demonstrate through simulation that in mobile environments, TES in conjunction with a simple link-layer loss avoidance scheme can gain an additional 20% improvement in aggregate throughput.

While achieving significantly high efficiency than DCF in many realistic scenarios, TES also achieve a higher degree of fairness among competing nodes. Compared to existing approaches, TES improves both long-term and short-term unfairness in many cases. Furthermore, TES is capable of achieving weighted fairness. I.e., TES can provide a channel occupancy time allocation among competing entities according to various priorities or weights of entities.

To summarize, TES achieves high network efficiency with little unfairness in both long and short timescales— the dual attributes that have not been achieved simultaneously by existing distributed MAC protocols.

Time-based Regulator

TBR is a backward-compatible link-layer scheduler that runs at the AP, works in conjunction with DCF, and requires no modifications to clients nor to DCF [94]. TBR is appropriate for existing AP-based networks in which a backward-compatible implementation is highly desirable. However, unlike TES, TBR is not effective when nearby non-cooperative nodes fall under different administrative domains.

We implemented TBR in the HostAP [50] driver for Prism-based 802.11b wireless interfaces and evaluated it on a Linux testbed. Based on a series of experiments reflecting realistic scenarios, we find that TBR is effective in allocating channel time equally among clients in the long-term. The current implementation of TBR is only effective for *single-cell* environments, in which contention only happens among nodes

within a cell. A *cell* is identified by a single AP and its associated client nodes. We also suggest how TBR can be extended to *multi-cell* environments, in which nodes in multiple nearby-by cells contend for channel access.

1.4.3 Organization

The rest of this dissertation is organized as follows. In Chapter 2, we describe background and related work in the areas of distributed wireless channel access and fair resource allocation. In Chapter 3, we describe an analytical framework for evaluating the impact of various fairness notions on achieved throughput and delay. We also describe a game theoretic model to analyze the impact of rational competition in non-cooperative environments on achieved throughput. Using this model, we prove that both DCF and its enhanced cousin EDCF (for Enhanced Distributed Coordinating Function), which is currently being drafted as part of the 802.11e standard, can lead rational nodes to inefficient equilibria. We also conduct simulation runs to specify the necessary conditions under which DCF leads rational nodes to inefficient equilibria. Using the framework presented in Chapter 3, we quantitatively compare the advantages and disadvantages between various pairs of fairness notions in Chapter 4.

In Chapter 5, we describe TES in detail. The chapter also contains an analysis on the channel access mechanism of TES, and reports sets of simulations comparing the performance of TES against some existing solutions, including DCF, for many realistic scenarios.

Chapter 6 describes our centralized, backward-compatible solution, TBR. We also describe our detailed implementation on a Linux testbed and our evaluation using it.

Finally, in Chapter 7, we conclude this dissertation by summarizing our work and contributions, discussing the lessons learned and outlining some directions for future research.

Chapter 2

Background and Related Work

In this Chapter, we provide background information and describe related work. In Section 2.1, we discuss the well-established research areas of fair queuing and processor sharing, where many known fairness notions, analysis methods and practical fair scheduling mechanisms have been established previously. We then describe major methods of arbitrating wireless channel access among multiple contenders. In Section 2.2, we survey related work in the area of distributed channel access protocols. We discuss the main ideas used by existing protocols to provide fair and efficient allocations of channel capacity among competing nodes. When relevant, we also discuss how our work differs from existing approaches.

2.1 Background

Our work mainly deals with issues related to the bottom two layers of the network protocol stack: the physical layer and the link layer. The main purpose of the physical layer is to provide a virtual bit pipe for transmitting a sequence of bits between any pair of nodes joined directly by a physical communication channel. The main purpose of the link layer is to provide a communication channel, typically with a loss recovery mechanism, for transmitting frames using the unreliable bit pipe. The link layer can further be divided into the MAC sublayer, which arbitrates channel access among competing nodes, and the logical link or data link control layer, which is responsible for framing and reliable transmissions.

2.1.1 Communications in Unlicensed Frequencies

Government regulations in the U.S. and in many countries over the past decade allow networked devices to operate in unlicensed wireless frequency bands. Neither the users of these devices nor the operators of WLANs need to pay fees for using an unlicensed channel. There is about 80 MHz of bandwidth centered around 2.4 GHz allocated for unlicensed wireless communications in the U.S. and many other countries. 802.11b and 802.11g devices operate in this spectrum, which is divided into 3 non-orthogonal (non-overlapping) channels [2]. There are also unlicensed frequencies available around

915 MHz; this band is mostly used by short-range communication devices.

Recently, 200 MHz of additional bandwidth (from 5.15 GHz to 5.35 GHz) have been allocated for unlicensed communications. Today, 802.11a devices operate in this spectrum [1]. However, because of its limited range at higher data rates (requiring almost line-of-sight) 802.11a devices are not as widely deployed as 802.11g and 802.11b devices.

2.1.2 Rate Diversity is Prevalent

Nodes connected to 802.11 WLANs transfer data at a number of different rates or speeds for two major reasons: i) a sender can transmit at a lower data rate (using a more resilient modulation scheme) to reduce the channel bit error rate (BER) and ii) different members of the 802.11 family co-exist in the same frequency band; for example, an 802.11b node with the maximum speed of 11 Mbps may compete against an 802.11g node with the maximum speed of 54 Mbps. Such environments are called *rate diverse*.

Various modulation schemes provide robust communication over wireless channels, which, unlike wired networks, are often lossy. Furthermore, the signal strength and loss rate of indoor wireless channels vary widely, even for nodes that are equidistant from access points [61]. When the 802.11 MAC protocol detects a frame loss (because of the absence of an acknowledgment frame or *ack*), it continues retransmitting the frame until the maximum retry limit has been reached. However, such retransmissions are futile when the average signal strength at the receiver is consistently lower than the threshold required for successful frame reception. In this and many other situations, the sender can improve performance by transmitting at a lower data rate using a more resilient modulation scheme so that the channel bit error rate (BER) is reduced at the expense of higher frame transmission time [42, 51]. Vendors of APs and client cards implement automatic rate control schemes in which the sending stations adaptively change the data rate based on perceived channel conditions [22, 51, 23, 24]. Many cards also allow users to manually set the data rate.

The 802.11b standard defines four different data rates: 1, 2, 5.5 and 11 Mbps. The 802.11g standard defines 8 additional data rates ranging from 6 Mbps to 54 Mbps.

To investigate the prevalence of rate diversity, we collected traces of wireless network traffic at a one-day IRIS student workshop at MIT on 11 August 2003. There were about 45 attendees and more than half turned on their wireless laptops. We set up a system to sniff data during two 90-minute workshop sessions, WS-1 and WS-2, which took place in a single room of about $12.2m \times 7.6m$. All nodes and APs were 802.11b devices. Figure 2-1 shows the fractions of data bytes transferred using each of the four possible rates during each session. Despite the relatively small room, during WS-2, more than 30% of the data bytes were transferred using data rates lower than 11 Mbps.

We also set up an experiment to investigate how an AP adapts data transmission rates to clients at different locations in indoor office environments. We placed a Cabletron Roamabout-2000 AP 2.2 m above floor in a office. A sender with a wired connection to the AP sent unicast UDP data packets at the saturation rate simulta-

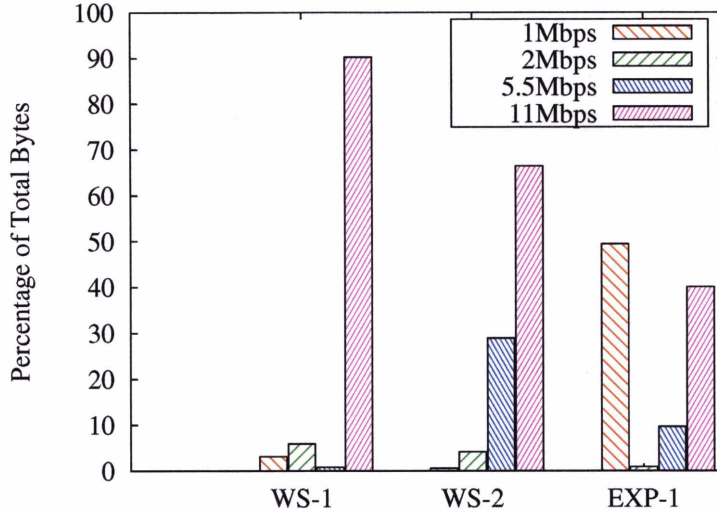


Figure 2-1: Fractions of bytes transferred at various data rates during three 90-minute student workshop sessions (WS) at MIT and an experiment (EXP-1)

neously to four different wireless receivers. The first receiver was about $1.2m$ away from the AP, the second $3.7m$ and one thin, wooden wall away, the third $7.9m$ and two thin wooden walls away and the fourth $9.1m$ and two thick walls away. As shown in Figure 2-1 (see EXP-1), more than 50% of the bytes were transferred using the lowest data rate. The AP used the lowest data rate mainly to transfer frames to the two most distant nodes.

2.1.3 Evidence of Multiple Users During Congested Periods

In the previous subsection, we introduced wireless network traces collected in a conference room at a student workshop held at MIT and showed that competing nodes connected to 802.11 WLANs often transfer data at a variety of data rates. Our analysis of this particular workshop trace data, however, showed that the network was well over-provisioned with 7 APs.

To provide evidence of many nodes competing for channel access during congested periods, we analyzed a wireless tcpdump trace of Whittemore, a residential facility in the Dartmouth business school where students were required to own laptops. This data was collected by Kotz *et al.* over the 2002 Spring semester [60]. Unfortunately, the trace data does not contain the data transmission rate used for each frame transmission. Nonetheless, we can identify the busy periods in which an 802.11b AP is carrying nearly the maximum amount of data, and investigate whether more than one user actively exchanged data during congested periods.

Because the trace data shows that TCP dominates this trace, we conservatively define *busy or congested intervals* as those in which the total data throughput at the

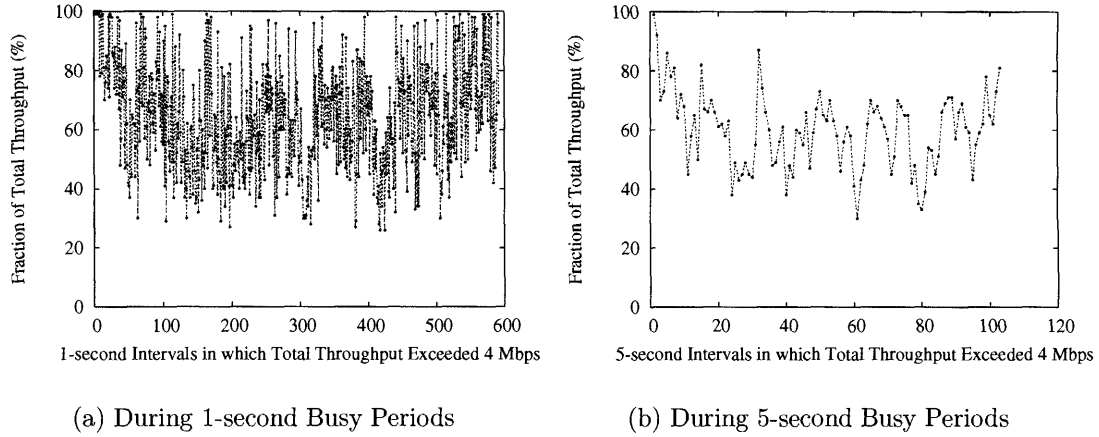


Figure 2-2: Fraction of throughput achieved by the heaviest user during busy intervals at an AP at a Dartmouth dormitory.

AP exceeded 4 Mbps, 80% of the commonly observed TCP saturation throughput of 5 Mbps when all competing nodes transmit at the maximum 802.11b data rate of 11 Mbps and experience a low loss rate of around 3%. This underestimates the congestion, because when nodes transmit at lower data rates, a considerably lower aggregate throughput is indicative of congestion.

Figures 2-2(a) and 2-2(b) plot the fraction of aggregate throughput achieved during busy 1-second and 5-second intervals by the heaviest user at an AP at Whittemore on Monday, 8 April 2002. The *heaviest user* is that node that exchanged the most bytes with the AP. The figures show that the heaviest user alone rarely saturated the channel. In most busy intervals, users other than the heaviest user exchanged significant amounts of data. This shows that busy periods stem from competition among multiple nodes.

As WLANs become ever more popular, many APs will potentially experience more congested periods with many nodes competing for channel access. Improving aggregate user utilities during those busy periods will become ever more important.

2.1.4 Loss Avoidance at the Physical Layer

Today's physical layers employ different modulation techniques that are robust for various channel conditions. Generally, different modulation techniques can lead to different transmission speeds and modulation schemes at lower transmission speeds are more resilient when the channel bit-error rate is high.

Forward error correction (FEC) techniques can also be applied in conjunction with modulation techniques. FEC is accomplished by adding redundancy to data bits. FEC techniques are most effective when channel errors are random. Furthermore, the effectiveness of an FEC technique depends on the channel bit error rate. The 802.11b

physical layer specification does not specify any FEC technique [2] but the 802.11a physical layer specification outlines various transmission speeds, each specified as a function of a modulation technique and an FEC level [1].

Despite various physical layer error prevention and correction techniques, frame losses can still occur because channel conditions vary dynamically even over short-time scales especially in indoor environments [64, 71, 83]. When the bit-error rate experienced by a sender varies because of time-varying and location-dependent channel conditions, it is challenging to compute an optimal combination of modulation and FEC for every frame transmission. The intelligence to select an optimal physical layer transmission strategy is often implemented at the link-layer, as discussed in the next section.

Furthermore, achieving a low or zero frame loss rate through physical layer error prevention and correction techniques does not necessarily lead to the optimal throughput. Under certain conditions, error recovery through link-layer retransmissions can lead to higher throughput than using a physical layer transmission technique that yields a low or zero frame loss rate. For instance, it is better to transmit at 2 Mbps and use a link-layer retransmission mechanism when the frame loss rate is less than 50% than to transmit loss-free at 1 Mbps.

2.1.5 Loss Detection, Avoidance and Recovery at Link Layer

Error Detection

The link layer attempts to provide reliable frame transmissions over an error-prone channel. An error detection mechanism and a retransmission technique are commonly employed to recover frame losses [8]. Today, cyclic redundancy check (CRC) codes are widely used for error detection in both wired and wireless channels. A typical error detection mechanism involves computing the CRC over data bits in each frame and including these bits in the frame (usually in the frame header or trailer) for the receiver to verify.

Retransmission Techniques

When a receiver cannot recover the received frame because of errors, retransmissions are often useful. Automatic Repeat Request (ARQ) is a commonly used retransmission technique under which the receiver requests (either explicitly or implicitly) retransmissions from the receiver.

The simplest ARQ technique is *stop-and-wait*. A stop-and-wait protocol ensures that each frame has been received correctly by the receiver before the sender transmits another (new) data frame [2, 5, 8, 74, 51]. Specifically, the sender transmits a data frame to the receiver and waits for an acknowledgment frame back from the receiver. If the receiver receives the frame successfully, it informs the sender by replying with a positive acknowledgment (*ack*). Otherwise, the receiver can either reply with a negative acknowledgment (*nack*) or not send any acknowledgment at all. Typically, both *ack* and *nack* frames include CRC bits as well. Only upon receiving an *ack*,

does the sender transmit the next data frame. If the sender has not received any *ack* or *nack* within a timeout interval, it retransmits the unack'd frame. The sender continues retransmitting the frame until an *ack* is received or the maximum retry threshold is reached, when it discards the frame. All current 802.11 variants use a stop-and-wait protocol. There are other possible ARQ techniques such as Selective Repeat that sends a cumulative acknowledgment requesting retransmissions of one or more packets that are not correctly received, instead of sending an acknowledgment for each frame being received. Such a protocol is more efficient than stop-and-wait [8].

FEC Techniques

Forward error correction (FEC) techniques can also be employed at the link-layer in the absence of an ARQ mechanism [53] or in conjunction with it [5, 21, 20, 33, 77]. Many existing techniques are found to be effective under some conditions even when the physical layer employs (different) FEC techniques [20, 33, 77].

Frame Size and Rate Adaptation

When the channel bit error rate is high, an adaptive protocol that chooses an appropriate frame size and/or transmission rate can improve achieved throughput. Transmitting large frames can increase the frame loss probability, requiring retransmissions, whereas transmitting small frames can decrease effective throughput due to the fixed per-frame overhead. Modiano [73] and Mitlin [69, 70] describe adaptive algorithms that attempt to compute the packet size yielding the maximum achieved throughput dynamically based on estimates of the channel bit error rate.

Similarly, transmitting at a lower data rate by using a more resilient modulation scheme leads to higher frame transmission time but generally reduces the frame loss rate. Although various transmission rates are provided by the physical layer, rate adaption schemes are typically implemented at the link layer for two reasons. First, selecting which rate to use typically requires estimates of channel conditions; link-layer retransmission and other signaling mechanisms can provide the feedback needed to estimate channel conditions. Second, the physical layer is often implemented in proprietary hardware or firmware and thus is often inaccessible. On the other hand, the link layer is often implemented as a software driver.

Kamerman *et al.* describes Automatic Rate Fallback (ARF) [51], a link-layer scheme that dynamically computes an appropriate data transmission rate for each frame transmission based on observed channel conditions. ARF employs a passive approach that only relies on the observed frame loss rate at each sender to improve achieved throughputs in WLANs based on WaveLAN II, a predecessor to the 802.11 technologies.

Receiver-based Auto Rate (RBAR) protocol [42] is another rate adaptation protocol designed to improve achieved throughputs in WLANs. Unlike ARF, RBAR employs an active approach that relies on feedback (the received signal strength) from the receiver to improve achieved throughputs in 802.11-based WLANs. The 802.11's RTS/CTS mechanism is used to receive immediate feedback from the re-

ceiver. Compared to ARF, RBAR is more responsive to dynamic channel conditions, but requires the use of RTS/CTS, leading to increased overhead. We explain the RTS/CTS mechanism in Section 2.1.8.

Opportunistic Rate Adaptation scheme (OAR) [87] builds on RBAR but achieves significant throughput gains in many cases. The key idea behind OAR is to allow nodes that have high-quality channel condition to transmit more than one packet at a time, taking advantage of time-correlated channel conditions. This idea is based on the observation that the channel coherence time, the average time of decorrelation between SNR values, spans multiple packet transmission times. Therefore, opportunistic scheduling policies can be employed to take advantage of good channel condition that on average remain for several packet transmission times. By allowing nodes with good channel conditions to transmit more packets than otherwise allowed under DCF, OAR also changes the definition of link-layer fairness, something we discuss fully in Section 2.2.4.

Loss Avoidance Through Spatial and Time Diversity

Another interesting approach to improve achieved throughput is to use more than one transmission path (i.e., *spatial diversity*) to take advantage of often independent channel conditions over transmission paths. Wireless channel errors are location-dependent. Miu *et al.* observe that in a typical WLAN deployment, there are areas where multiple APs provide overlapping coverage, leading to a possibility of transmitting to a client from different APs on a per-frame basis [71]. Using experimental measurements and analysis, they show that i) losses are bursty on each transmission path, identified by a transmitter-receiver pair, and ii) the loss statistics along different paths often demonstrate little temporal correlation, especially when the receiver is moving. They propose a fine-grained path selection scheme called Divert that selects the best AP for down-link communication in a way that ensures that the likelihood of success of the next frame following a lost one is higher than if the current AP were used.

A similar idea was used in MRD [72], which attempts to avoid time-correlated frame losses through multiple frame receptions, each from a different radio. Specifically, multiple radios connected to a wired infrastructure forward their (possibly corrupted) copies of a frame to a centralized coordinator, which then attempts to recover the frame by combining those copies.

Both of these approaches take advantage of *spatial diversity* or *antenna diversity* at the link layer. There are also physical layer techniques that take advantage of antenna diversity.

Several schemes avoid losses by dictating “when to transmit” frames [62, 81]. This is because when the channel conditions are time-varying, it is more efficient to transmit when the channel is in a “good state” than when it is in a “bad state.” Link scheduling schemes that take advantage of *time diversity* are known as *channel state scheduling* schemes.

The authors of [62, 81] propose a rate adaptation scheme in which senders defer transmissions (i.e. give up their transmission opportunities) when they are observing

bad channel states. Meanwhile, other nearby senders who are experiencing better channel conditions can transmit, thereby improving the overall efficiency. A MAC protocol should encourage rational nodes to employ such transmission strategies to improve the overall efficiency. However, when a node frequently defers its transmissions voluntarily, it is likely that a lesser share of channel capacity is allocated to it in the long-term than what it would achieve if it transmitted whenever it had the chance to do so. Thus, the deployment of such a scheme can lead to unfair allocations of channel capacity. We discuss more on this in Section 2.2.5.

2.1.6 Fixed Assignment Techniques for Channel Access

Channel access protocols are needed to arbitrate channel access when there are multiple competing entities. The goal is to provide a fair division of channel capacity and to reduce *collisions*, which occur when multiple frame transmissions from two or more nodes overlap. The simplest way to divide channel capacity among competing entities is through a static assignment of time, frequency or code. Under time division multiple access (TDMA), users are assigned fixed time slots. Only one user is active in each time slot and the designated user can use the entire channel bandwidth during its time slot. Under frequency division multiple access (FDMA), each user is statically assigned to a fraction of the available channel bandwidth and not allowed to use the rest of the channel bandwidth. However, both techniques lead to wasted channel capacity and increased delay when user traffic is highly bursty, which is the main characteristic of data networks [8, 100].

Under code division multiple access (CDMA), different signaling codes are used for different transmitters, allowing multiple transmissions to overlap both in frequency and time coordinates. Generally, multiple orthogonal codes are obtained at the expense of increased bandwidth requirements. Although CDMA can allow the coexistence of multiple nearby transmitters without explicit coordination, the transmit power of each sender transmitting to a common receiver (base station) must be controlled in a fine-grained manner so that senders that are closer to the base station will not drown out senders that are further away (the *near-far* problem) [83]. CDMA is used in cellular networks and the near-far problem is avoided by having each base station implement a complex power control scheme that provides feedback to each mobile client on how to adjust its transmit power.

Typically, TDMA, FDMA and CDMA techniques are employed in a coordinated fashion in which a centralized coordinator in each cell is responsible for coordinating and providing orthogonal channel access among competing nodes. All three techniques have been used in cellular telephony networks. Early satellite communications systems use TDMA and FDMA techniques.

In wireless data networks, traffic is bursty and coordinated access is often impractical. Therefore, in these networks, randomized, decentralized channel access protocols are typically used. They are the topic of discussion in the next section.

2.1.7 Distributed Channel Access Protocols

Aloha [3] is one of the earliest distributed, randomized channel access protocols. It was first developed for the Aloha satellite communication system. In Aloha, each node can transmit any time it wants. Upon transmission, the sender waits for an acknowledgment from the intended receiver. If the sender does not successfully receive an ack, it assumes that a collision has occurred and attempts to retransmit. Each retransmission attempt is preceded by a randomized amount of delay to avoid repeated collisions among retransmitting nodes.

A slotted version of Aloha (called Slotted Aloha) was developed later. In Slotted Aloha, each user transmits packets at a time slot boundary; each time slot equals the transmission time of a single frame (assuming all frames have the same length). Therefore, in Slotted Aloha, transmissions involved in a collision overlap completely instead of partially, increasing efficiency over pure Aloha.

Aloha-like protocols are also known as *contention-based* protocols. When there is a large population of bursty users and the total offered load is considerably less than the system's capacity, Aloha-like protocols, when compared to TDMA and FDMA, improve the average per-packet delay and bandwidth requirements to support the maximum number of users each with the same offered load [100]. However, under Aloha-like protocols, the collision rate increases rapidly as the offered load increases.

Carrier-sense multiple access (CSMA) protocols [55, 99] improve performance over Aloha by exploiting information about other users. In CSMA, before transmitting its frame, each user listens to the carrier frequency and transmits only if the channel is sensed idle. If the channel is sensed busy, the user waits until the channel is idle again and attempts to transmit in each subsequent idle slot with a certain probability. CSMA schemes differ mainly on how they choose the distribution of transmission probability and on how they compute the probability of transmission.

These *listen-before-talk* protocols effectively avoid many collisions that would have occurred under an Aloha-like protocol. Observe that in CSMA, when all users are within radio range, collisions can only occur if at least one user transmits before it can detect the existence of a preceding transmission of another user. The *maximum detection delay* is the sum of the maximum propagation delay and hardware detection delay. Unlike in Satellite communication systems where the propagation delay can be relatively large, in wireless local area networks, this delay is a few orders of magnitude smaller than the frame transmission delay. For example, in 802.11b, the detection delay is less than $10\mu s$ whereas the transmission delay of a 1500-byte frame at 11 Mbps is about $1.37ms$. Like Aloha, CSMA can be unslotted or (time) slotted. In CSMA, it is easier for senders to get synchronized at slot boundaries since each competing node will detect the end of a frame transmission (the beginning of idle channel) at roughly the same time.

Many CSMA-based distributed MAC protocols have been proposed in the past [9, 12, 13, 31, 32, 34, 40, 52]. We will discuss them in detail in Section 2.2. Our proposed MAC protocol, TES, is also a variant of CSMA. The next section describes 802.11's DCF, another variant of CSMA. Familiarity of the detailed operations of DCF is necessary so that we can compare it with other existing approaches as well as with

our new approach.

2.1.8 802.11's Distributed Coordinating Function

802.11's DCF (Distributed Coordinating Function) is the most commonly used contention resolution method in 802.11 networks. Although the 802.11b standard specifies an alternative centralized (poll-based) mechanism, Point Coordinating Function (PCF), the implementation of PCF is optional and is not implemented by most AP vendors because of its complexity and issues of co-existence with DCF-based networks.

Under DCF,

1. Each sending node i contends for a *transmission opportunity* (TXOP) as follows:
 - (a) Sets a backoff counter to a random integer from a uniform distribution between 0 and CW_i , the contention window of node i ,
 - (b) Decreases the counter by 1 for each time slot in which the channel is sensed idle,
 - (c) Pauses the counter for each busy time slot during which it senses carrier,
 - (d) Transmits a frame (a new frame or a retransmission) at the beginning of the idle slot when the backoff counter reaches 0,
2. Each receiving station, upon receiving a successful frame transmission, replies with a *synchronous acknowledgment* frame (*syn-ack*) after a pre-defined interval called SIFS (short inter-frame space), and
3. Each sending station i that has just transmitted a frame stops and waits for an acknowledgment frame:
 - (a) Upon receiving a successful acknowledgment frame, it resets CW_i to CW^{min} and goes to Step 1, and
 - (b) Upon not receiving a successful acknowledgment after a timeout, it increases CW_i exponentially by setting $CW_i \leftarrow \min((CW_i + 1) * 2, CW^{max})$ and goes to Step 1.

In any DCF-like CSMA MAC protocol, the contention window CW_i of node i dictates the allocation of the shared channel. Figure 2-3 illustrates the backoff mechanism when nodes A and B compete for channel access. When all nodes are within transmission range, a collision occurs if and only if two or more nodes choose the same number of backoff time slots. A time slot is $20\mu s$ in 802.11b and SIFS is $10\mu s$. Generally, the maximum detection delay must be less than SIFS. Notice that the backoff counter (see Step 1c) is paused whenever a node senses that the busy medium is busy (i.e., senses carrier) due to a successful frame transmission from another node or collided transmissions.

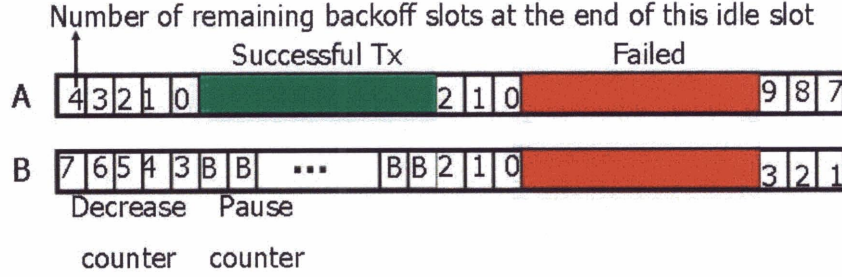


Figure 2-3: An example illustrating DCF's backoff counters of competing nodes A and B.

The CW_i of each node i is initially set to CW^{min} , which is 31 for 802.11b. In general, for a fixed CW value, the probability of collisions increases with the number of competing transmitters or contenders. Also, the larger the CW_i of each contender i , the smaller the probability of collisions. Therefore, it is important that the CW_i of each contender i increases with the number of contenders. However, it is difficult to estimate the number of contenders given the dynamic nature of traffic and varying sending rates at contenders. DCF adopts a simple exponential backoff approach in which each sender i multiplies the contention window size with 2 whenever it experiences a frame loss (see Step 3b). Thus, CW_i is doubled with each successive frame loss up to the maximum value $CW_{max} = 1024$. The goal is to double the average backoff interval ($\frac{CW_i}{2}$) with each successive frame loss. However, since the minimum number of backoff slot can be zero, in practice, DCF uses the following formula, a more complicated formula than the one showed in Step 3b, to update CW_i to achieve the goal of doubling average backoff interval: $CW_i \leftarrow \min(2 * (CW_i + 1) - 1, CW^{max})$.

As shown in Step 1, CW_i is set to CW^{min} as soon as the sender is able to successfully transmit a data frame, as indicated by successful reception of the syn-ack. There are advantages and disadvantages to DCF's backoff mechanism. By setting $CW_i = CW^{min}$, DCF quickly reduces the expected amount of idle time between transmissions, increasing aggregate throughput when contention is low. However, this also has the effect of increasing the collision rate when the number of contenders is more than a few, as shown in Section 1.2.4.

DCF also includes a *collision avoidance* mechanism (thus making it a CSMA/CA protocol) called RTS/CTS (for request-to-transmit/clear-to-transmit). The RTS/CTS mechanism is an optional mechanism that can be used to minimize *asynchronous collisions* caused by the hidden terminal problem. As explained earlier, when all competing nodes are within radio range, CSMA ensures that only *synchronized collisions* happen, i.e., all frame transmissions involved in a collision were transmitted in the same time slot. However, the hidden terminals can lead to asynchronous collisions, in which one or more frame transmissions collide with an existing frame transmission that was transmitted more than one time slot ago. We illustrate the hidden terminal problem with a simple example. Consider three nodes, A, B and C, in a chain configuration. Assume that A and B are within radio range and B and C are also within radio range but A and C are not. While A is transmitting a frame to B, C

cannot hear A 's transmission and wrongly believes that the channel is idle (at B). If C transmits, a collision will ensue at the receiver B . Therefore, the listen-before-talk operation becomes ineffective in the presence of hidden terminals.

The hidden terminal problem cannot be eliminated entirely in CSMA. However, its adverse effects can be significantly reduced by *floor acquisition* mechanisms such as 802.11's RTS/CTS, MACA [52], MACAW [9] and FAMA [31, 32]. In DCF, when RTS/CTS is enabled, each competing sender will send a short control frame, RTS, to its receiver. Upon receiving an RTS frame, the receiver will reply with a CTS frame if the channel is sensed idle by the receiver. Upon successfully receiving a CTS frame, the sender can transmit a data frame. Each nearby node that hears a RTS or CTS frame will suspend its transmission for the duration specified in the control frame. In the previous example, if B receives an RTS frame from A , it will reply with a CTS frame, which will also be heard by C , which then suppresses its own transmission. Therefore, in this case, RTS/CTS is effective in preventing asynchronous collisions of data frames. Asynchronous collisions of RTS frames can still happen just as with data frames when RTS/CTS is not used. However, RTS frames are much smaller than data frames leading to a smaller loss of aggregate throughput due to collisions.

Although RTS and CTS frames are small in size, in 802.11b, due to a relatively large fixed per-frame overhead, the transmission time of RTS/CTS frames can be as much as 20% of the transmission time of a data frame at 11 Mbps. Therefore, it may not be a good idea to always enable the RTS/CTS mechanism. How best to employ RTS/CTS or more precisely how to adaptively enable and disable the RTS/CTS mechanism is still an open question.

Another problem with RTS/CTS is that it can lead to the exposed terminal problem, in which two transmitters are kept from transmitting to corresponding receivers even though doing so cannot lead to collisions at both receivers. This is because collisions happen at the receivers and RTS/CTS prevents nearby transmitters from transmitting even though their transmissions cannot lead to collisions at the receivers.

2.1.9 Fair Queuing

The problem of fair channel capacity allocation is related to fair queuing which has been an active area of networking research for decades [16, 25, 37, 67, 90, 91]. The main goal of any fair queuing scheme is to provide fair allocations of bandwidth at network routers, switches and gateways without sacrificing performance.

A major motivation behind fair queuing is to prevent ill-behaved sources from getting larger shares of bandwidth by sending packets in a greedy manner. When fair queuing schemes are implemented at the routers, these ill-behaved sources will be limited to their fair shares.

Allocation of bandwidth is often done on the basis of network-layer source-destination pairs [25]. That is IP source-destination pairs are considered competing entities.

Most fair queuing algorithms attempt to achieve a *max-min* fairness criterion for shared entities with equal priorities. Under max-min fairness, an allocation is fair if i) no entity receives more than its request ii) no other allocation scheme satisfying condition i) has a higher minimum allocation and iii) condition ii) remains true recursively

as we remove the entity with the minimal allocation and reduce the total resource accordingly [25, 47]. The max-min fairness criterion has been widely accepted in the networking community.

In general, fair queuing schemes require multiple queues, each of which holds packets from an entity or a group of entities. A simple round-robin scheduling mechanism that schedules packets from these queues in a round-robin fashion can achieve fair allocations of packets sent. However, packets vary in sizes and thus pure round-robin scheduling fails to achieve fair allocations of bits. It is widely accepted that transmitted bits are “fairer” notions of fairness than transmitted packets.

In wired networks, packets from different flows that share an outgoing link are transmitted at the same transmission speed. Since there is no rate diversity, the channel capacity is independent of how packets are scheduled as long as the scheduling scheme is work conserving. As we discuss throughout this thesis, in the presence of rate diversity, the wireless channel capacity can vary depending on how frame transmissions are scheduled and thus channel occupancy time not bits should be the fundamental allocation unit.

Fair queuing is also applicable when competing entities have varying priorities or weights. Formally, for each entity i , let w_i be the fair share of entity i and $A_i(t_1, t_2)$ be the number of allocation units allocated to entity i between interval $[t_1, t_2]$. The goal of a fair queuing algorithm is to minimize $|\frac{A_i(t_1, t_2)}{w_i} - \frac{A_j(t_1, t_2)}{w_j}|$ for any pair of entities i and j over an arbitrary interval $[t_1, t_2]$ in which both entities are backlogged. Weighted Fair Queuing (WFQ) [25], Self-Clocked Fair Queuing (SCFQ) [36] and Packet Generalized Processor Sharing (PGPS) [78] are examples of packet fair queuing algorithms that are generally effective in achieving weighted bit-based fairness (at network queues). In general, these schemes achieve weighted fairness in the following manner. First, they compute the finish time of each packet of a flow upon entering the queue designated for the flow. The computation is a function of the size of the packet, the minimum of the finish time of the last packet in the queue and the current time, and the fair bandwidth share of the flow. Second, they schedule head-of-line packets from flow queues according to a non-decreasing order of finish time. There are also schemes such as Deficit Round Robin [90] that use more efficient implementation techniques to provide weighted fairness.

Adapting wireline fair queuing schemes to the wireless domain is challenging for the following reasons:

- Competing nodes often experience varying channel conditions; how best to divide allocation units fairly in such cases is unclear,
- There is no centralized coordinator that has full knowledge of channel conditions and backlogged queue information of each wireless node,
- Nearby nodes often fall under different administrative domains and thus centralized fair scheduling is often impractical, and
- A different fairness notion other than bit-based fairness may be desirable to encourage efficient use of the shared channel among nodes that may transmit

at different data rates.

Much interesting work has been done in the area of distributed wireless fair scheduling for the past decade and we discuss some of this work in the next section.

2.2 Related Work

For the remainder of this chapter we focus on issues related to providing fair and efficient capacity allocation among multiple competing nodes using CSMA-based channel access protocols. For a comprehensive treatment of wireless channel access protocols and other issues that we are not covering in this section, see [8, 83, 86].

2.2.1 Collision Detection

In a wired shared medium, detecting collisions is possible by having a sender listen to the channel while transmitting (*listen-while-talk*). When the sender cannot decode its own transmission, it assumes that a collision has happened and immediately stops its transmission, and then delays for a random amount of time before transmitting again. Ethernet [68] is a popular example of CSMA/CD, and it has been shown that its theoretical channel utilization can approach 100% when the detection delay is relatively much smaller than the transmission delay [8]. As the transmission delay gets smaller, because of faster data rate or smaller frame size, CSMA/CD becomes less efficient.

However, the same concept is not applicable in wireless networks for three practical reasons:

- Most existing wireless systems are half-duplex systems, that use a single radio for both transmitting and receiving. In such systems, collision detection must be done by interrupting the transmission to sense the channel. A full-duplex transceiver or multiple radios is required to avoid interruptions during transmissions,
- Even with a full-duplex radio, each sender may not be able to accurately detect whether a collision has happened just by listening because collisions happen at the receivers (not at the senders), i.e., even though the receiver does not successfully receive the frame, the sender might think otherwise,
- Wireless channel bit-error rates (e.g., 10^{-5}) are typically several orders of magnitude higher than bit-error rates of wired links (e.g., 10^{-13}). Therefore, if the transmitter cannot decode its own transmission, it may be because of channel errors and not collisions.

For these reasons, CSMA/CD is not employed in today's wireless LAN technologies. Instead, collision resolution or avoidance mechanisms are used to mitigate the adverse effects of collisions.

2.2.2 Collision Resolution

Both Aloha and CSMA are shown to be unstable [100, 8], i.e, the departure rate of the system decreased with increased load, leading to growing backlogs of frames that will not be successfully transmitted by nodes within a finite interval. The stability analysis often includes an assumption of infinite set of nodes and thus provides an upper bound to the delay achieved (in practice) with a finite set of nodes. In any case, in both Aloha and CSMA, the aggregate throughput decreases with increased load because of the increased collision rate.

To achieve stability as well as to improve achieved throughput, many collision resolution mechanisms have been proposed in the past [52, 9]. All these mechanisms rely on the following simplifying assumption: the sender has a way of knowing that a collision has occurred a small amount of time after the frame transmission is completed. Most protocols use the link-layer ARQ mechanism to achieve that goal. That is if a sender has not successfully received an *ack* from its intended receiver after a timeout, it assumes that a collision has occurred.

Splitting algorithms [17, 101] were initially proposed to stabilize Aloha. These algorithms divide nodes into two sets whenever a collision occurs: one with all the nodes not involved in the collision and the other with those involved in the collision. Nodes in the first subset transmit in the next time slot. And, if the next time slot is sensed idle, those in the second subset transmit in the subsequent idle slot. Nodes in the second subset always wait until any collisions that occurred during transmissions from the first set of nodes are successfully resolved. This procedure continues recursively until all collisions are resolved.

Stabilization mechanisms using splitting techniques and other adaptive techniques were also developed for CSMA [8, 34]. FAMA-CR [34] is a CSMA protocol that employs a splitting approach. Under pseudo-bayesian approaches, at the end of each time slot, each node estimates the number of contenders dynamically based on whether the time slot was idle, whether the frame transmission was successful or whether it has involved in a collision [8]. These protocols are shown to be effective so long as there exists an effective feedback mechanism that informs all competing nodes of whether a frame transmission is successful or failed because of collision (and not because of channel errors).

Bounded binary exponential backoff mechanisms are used in MACAW [9], Ethernet [68] and DCF to reduce the collision rate as the number of contenders increases. Such mechanisms can lead to instability depending on the details of the implementation. For example, DCF's exponential backoff mechanism leads to significantly low aggregate throughputs when the number of backlogged contenders increases beyond a dozen or so nodes [12, 97].

Exponential backoff mechanisms and many other approaches use a uniform distribution of transmission probability. Hiperlan [41], Sift [48] and CSMA/ p^* [98] use non-uniform distribution of transmission probability. Sift and CSMA/ p^* are shown to resolve collisions quickly in event-driven workloads where many nodes simultaneously attempt to send data at the time of an event (this type of workload can be found in sensor networks).

The main problem with all of these approaches is the implicit assumption that a frame loss is *always* because of collisions. Clearly, this is not true in a typical indoor wireless environment where the rate of frames being lost because of channel errors can well exceed the collision rate. Therefore, these protocols can lead to degraded throughput. Furthermore, when the bit error rate experienced by each competing node varies, as they often do [71, 83], unfair allocations of channel capacity ensue.

We propose that each competing node adjusts its contention window size based on its observed idle time between transmission events. Through an analysis, we show that i) there exists an optimal amount of idle time between transmission events (for a given average transmission time), that maximizes aggregate throughput and ii) the optimal amount of idle time between transmission events varies little with the number of contenders. Thus, the optimal amount of idle time can be pre-computed and each contender can adjust its contention window size dynamically so that the observed idle time equals the target idle time. A sender increases(decreases) its contention window size when the observed idle time between transmission events is smaller(larger) than the target idle time. In Chapter 5, we explain in detail this contention resolution mechanism, a major component of our TES protocol.

Heusse *et al.*, a channel access protocol called *Idle Sense (IDS)* that adjusts contention window size based on observed idle time [40]. IDS and TES share some of the core observations and ideas on using idle time to resolve contention. Bononi *et al.* make similar observations about idle time in [13] and propose Asymptotic Optimal Backoff (AOB), a mechanism that extends the exponential backoff mechanism by adjusting the transmission probability of each node through channel time slot utilization and the average frame transmission time. However, unlike TES and IDS, AOB still retains the 802.11 exponential backoff mechanism and the number of retransmission attempts (as an indication of congestion) is still used in computing the transmission probability. Therefore, AOB is effectively a feedback-based approach like DCF. Unlike AOB, DCF and other feedback-based approaches, both IDS and TES decouple collision resolution from the link layer's mechanism to achieve reliability. Compared to DCF, both TES and IDS reduces collision rates significantly, thereby improving aggregate throughput especially when the number of contenders is more than a dozen.

TES differs from IDS in the way it adjusts the CW of each node as a function of observed and desired amounts of idle time. TES and IDS lead to comparable aggregate throughput only in the long-term and in the absence of time-correlated channel errors. TES achieves higher aggregate throughput than IDS on short-time scales when the number of contenders is more than a few dozen. In the presence of time-correlated channel errors, TES achieves significantly higher throughput than IDS. IDS leads to aggregate throughput that is even lower than DCF when loss rates because of channel errors are high. This is because TES, unlike IDS, employs a simple burst loss avoidance mechanism.

Furthermore, TES has a fairness controller, which IDS does not. Unlike IDS, TES: i) provides long-term time share guarantees among competing links, even in the presence of burst loss avoidance mechanisms, thereby leading rational nodes to employ efficient burst loss avoidance schemes, and ii) achieves arbitrarily weighted time-based fairness. We compare TES, DCF and IDS through simulation in Chapter 5.

2.2.3 Distributed Weighted Fair Scheduling

Efforts have been made in developing distributed fair scheduling algorithms that are suitable for the shared wireless medium. Like the schemes proposed in wired networks [16, 25, 37, 67, 90, 91], these wireless scheduling algorithms [29, 64, 82, 102] do not take into account the impact of transmission rate diversity. For simplicity of discussion in this subsection, we assume that competing entities are nodes, which is the assumption of most protocols.

DFS (for Distributed Fair Scheduling) [102] emulates SCFQ (for Self-clocked Fair Queuing) [36] in a distributed fashion. DFS assumes that for each node i , its fair share of the channel capacity ϕ_i is known. DFS achieves weighted fairness by having each node select a contention window size that is inversely proportional to its fair share of transmission opportunities. However, for a particular node i with a small value of ϕ_i , the contention window size can be quite large; this in turn results in a large amount of idle time between transmission events, leading to long packet latency and reduced throughput. To solve this problem, each node running DFS computes the contention window size using an exponential scaling factor and whenever it senses a frame transmission from another node, it re-computes its backoff interval based on the backoff interval included in that frame's header. Clearly, this mechanism will not work when competing nodes often cannot decode the headers of each other's transmissions. This can occur frequently in AP-based WLANs. For instance, imagine there are two clients associated with an AP; each client can exchange data with the AP and their frames can collide at the AP, yet they may not be able to successfully receive each other's transmissions. Furthermore, their evaluation shows that the aggregate throughput under DFS can be lower than that under DCF in several cases.

Unlike DFS, TES decouples the mechanism to achieve fairness from that to achieve efficiency. Each node running TES attempts to maximize the overall efficiency by observing idle time between transmission events without requiring competing nodes to signal or exchange information. TES's fairness controller works independently from its efficiency mechanism. Each node running TES observes the actual amount of channel time it uses and the amount used by all other nodes. Using this information and the node's assigned fair share, TES achieves long-term weighted fairness among competing entities.

Lu *et al.* methodically investigate issues concerning providing fair scheduling and QoS in AP-based networks with CSMA MACs [64]. They propose WFS (for wireless fair service), which includes a centralized coordinator running at the AP that allocates bit-based fairness among flows according to statically assigned rate weights. In WFS, each client informs the AP of its backlogged queue status. They also show that WFS can be implemented (with modifications) within the framework of a CSMA MAC such as DCF. Their protocol only works in single-cell environments, since coordination among competing nodes through the AP is necessary.

In contrast, TES works in both single-cell and multi-cell environments and no coordination among the APs nor among the APs and clients is necessary. As a result of WFS's explicit and careful coordination between the AP and the clients, WFS can provide a higher degree of short-term fairness than can TES.

Fang *et al.* proposes Estimation-based Fair Medium Access (EFMA), an interesting approach to achieving weighted fairness among competing nodes in a distributed fashion [29]. In EFMA, each node observes every transmission and maintains two variables: the total amount of time used by it to transmit frames and that used by other nodes to transmit their frames. Based, on these two variables, EFMA computes an index that measures whether a node is achieving its fair share or not. Each node uses this index to update its contention window size in a MIMD (multiplicative increase multiplicative decrease) manner. Like DFS, a major limiting factor of EFMA is that in many cases, the aggregate throughput is noticeably lower than DCF.

TES provides weighted fairness like DFS does but unlike DFS, increases aggregate throughput over DCF.

2.2.4 Distributed Time-based Fair Scheduling

The general idea of temporal sharing in the context of multi-rate WLANs is discussed in [87]. They propose a rate adaptation scheme, OAR, nodes that have high-quality channel conditions to transmit more than one packet at a time, taking advantage of time-correlated channel conditions. OAR dictates that the number of frames that each node transmits in each transmission opportunity is the floor of the ratio of its data rate and the base data rate (e.g., 2 Mbps). Therefore, a node transmitting at 11 Mbps can send 5 frames back-to-back and a node using 2 Mbps can only send 1 frame. Their approach leads to a *rate-based fairness* notion, which is a *crude* but not accurate approximation of time-based fairness. Furthermore, their approach is a special case of a generalized time-limited-TXOP approach, the drawbacks of which we will discuss shortly. OAR is a DCF-based protocol mainly intended for *ad hoc* networks and requires modifications to DCF. The paper did not explore the impact of time-based fairness (irrespective of channel conditions) on the achieved throughput and delay of each user.

In this thesis, we reconsider various notions of fairness in WLANs according to their choice of entities and *fairness units*. We investigate their impact on achieved throughput and delay through quantitative and trace-driven analyses, experiments using a test-bed, and simulation. Unlike previous work, we present a precise definition of time-based fairness under which each competing entity is allocated an equal share of channel occupancy time and argue that a MAC protocol should provide time-based fairness. Our proposed TES MAC protocol achieves time-based fairness in the presence of varying data rates, frame sizes and channel errors.

There are two straightforward ways to provide time-based fairness among competing nodes with equal priorities. First, each node is allowed to transmit one frame at a time but the number of transmission opportunities granted to each node varies according to the data rate used. Second, the transmission opportunities can be time-limited instead of size limited and each node is granted an equal number of transmission opportunities.

Under the first approach, each node is only allowed to transmit one frame at a time but the contention window of each node is scaled according to its transmission rate. For example, for any contention window based MAC protocol, the contention

window CW_i of node i can be updated after it is computed according to its backoff mechanism as follows: $CW_i \leftarrow CW_i \times \frac{r_{max}}{r_i}$, where r_{max} is the maximum data rate allowed in the system and r_i is the data rate used by i . This type of approach is suggested in [40]. Such a static scaling technique is problematic for two reasons: i) r_i may vary for each frame transmission when auto rate protocols are used to improve efficiency and ii) restricting each node to transmit only one frame in each TXOP is not efficient, especially when the transmission rate is high.

Under the second approach, the transmission opportunities are time-limited. That is upon winning a TXOP, each node can transmit a number of frames consecutively up-to a system-wide pre-configured time limit called TXOPLimit. For example, when TXOPLimit is set to 12.5 ms, a node can transmit a single frame at 1 Mbps or a *burst* of 9 frames (consecutively only separated by a small amount of time and/or *acks*) at 11 Mbps. Under this approach, a MAC protocol only needs to ensure that each node gets an equal number of transmission opportunities. This is the approach taken by EDCF, as specified in the 802.11e draft specification [44], and already implemented by some 802.11 card vendors [26, 75]. The time-limited-TXOP approach can significantly improve performance especially at faster transmission rates since the overhead per transmission event is reduced. However, TXOPLimit cannot be arbitrarily large since doing so can lead to significantly increased per-frame delay, adversely affecting delay sensitive applications as well as TCP flows because of TCP timeouts.

The time-limited-TXOP approach can provide time-based fairness among competing backlogged nodes only if all nodes transmit up-to TXOPLimit in each TXOP or, more generally, if the average duration of transmission burst of each node is the same. However, the optimal duration of transmission burst may widely vary among nodes (i.e., a system-wide TXOPLimit is not optimal for all nodes) because the channel conditions of each node can drastically vary. We give two examples. First, assume that the transmission rate used by a node is low such that it can only transmit at most one frame during TXOPLimit. As demonstrated in earlier work [69, 70, 73], the optimal frame size varies with the channel bit error rate. Therefore, it may be more efficient for a node to use smaller frames instead of the maximum sized frames allowed under TXOPLimit.

Second, when a node is allowed to transmit multiple frames in each TXOP, it should avoid wasting capacity by stopping its transmissions when it detects that the earlier frame transmissions have failed. This way, a node avoids transmitting frames that are highly likely to fail when it is i) experiencing time-correlated channel errors and ii) involved in a collision. The detection mechanism can be implemented by having the receiver reply with an *ack* after receiving each data frame in the transmission burst. The next section discusses fairness related issues in the presence of diverse channel bit error rates.

2.2.5 Fair Share Guarantees and Compensation

A fundamental issue related to the previous discussion on fairness is how best to provide fairness in the presence of i) location-dependent errors and ii) time-correlated errors. DCF and most existing distributed protocols do not deal with these issues

[70, 29, 40, 82, 87, 102]. Under these protocols, fairness is achieved in a “best-effort” or *use-it-or-lose-it* manner. That is if a node does not transmit when it has a chance (i.e., when it wins the contention), it will forever lose its transmission opportunity. However, in the presence of time-correlated errors, it might be more efficient for a node to intentionally delay its transmission and to allow other nodes with better channel conditions to proceed, thereby improving the overall efficiency. However, a rational node will not give up its TXOP if it is going to lose it forever. We argue that a reasonable compensation mechanism should be implemented at the MAC layer so that each rational node will employ the most efficient transmission strategy, so that aggregate throughput is maximized with respect to a particular fairness notion.

Similarly, the most efficient transmission strategy for a node may be to transmit at a frame size smaller than the maximum size allowed but in the presence of competition, it may not do so if it does not receive any compensation for the lost transmission time for using smaller frames than other nodes. A compensation mechanism can encourage rational nodes to only use the most efficient transmission strategy.

In the context of centralized scheduling, Lu *et al.* explores the idea of compensating for flows that decline to send packets intentionally because of perceived poor channel conditions [64]. Under their approach, a link-layer centralized scheduling scheme at the AP attempts to avoid time-correlated errors by dictating frame transmissions at the AP and clients [64].

We agree with the authors that the link layer should provide a compensation mechanism. We provide a similar but more general notion of compensation, in which each competing entity is: i) allocated its fair channel time share in the long-term and ii) compensated in the future for “lost” channel time up-to a maximum pre-configured amount. In contrast to their centralized approach, we provide a distributed fairness mechanism as part of TES that provides long-term time share guarantees without requiring any explicit coordination. Our fairness mechanism is flexible and can work with any burst loss avoidance scheme, including DCF-like exponential backoff schemes. We describe TES’s distributed fairness mechanism in detail in Chapter 5 and show that it can provide fair time share guarantees as well as encourage rational nodes to employ efficient transmission strategies, thereby improving the overall network efficiency.

2.2.6 Throughput, Delay and Equilibrium Analyses

In this section, we discuss related work in the areas of analyzing throughput, delay and equilibrium analyses.

Throughput Analysis

The capacity of 802.11-based networks has been analyzed in [10, 15, 54, 97]. We build on the approximate throughput analysis in [97] and show that the amount of idle time between transmission events that maximizes the aggregate throughput is roughly independent of the number of contenders.

Recently, Heusse *et al.* showed through simulations and experiments that performance degradation occurs when two nodes are sending at different data rates [39]. Through analysis, the authors show that the node sending at a lower data rate will achieve the same throughput as other nodes sending at higher data rate, leading to the low aggregate throughput. The authors do not suggest any mechanism to mitigate this effects.

We make similar observation. But unlike previous work, we provide a simple, organized framework that allows us to conduct analysis on achieved throughput, session delay and equilibrium situations for different fairness notions. Using our analytical framework, we conduct analyses that clearly show the impact of various fairness notions including frame-based fairness and time-based fairness on achieved throughput and delay.

Delay Analysis

It is well known that a scheduling policy can significantly affect the delays experienced by tasks of various sizes. In the absence of job size information, many scheduling schemes that do not use task size information such as First-in-First-Out (FIFO) and Last-in-First-Out (LIFO) can lead to high average delays. The main problem with any of these schemes is that the small job that arrives behind large jobs has to wait a long time to receive the service (generally known as the *slow truck effect*). The average delay under such a scheduling scheme increases with the variation in service time, more precisely with the second moment of the service time [56].

The average delay under processor sharing does not depend on the service time distribution. Under the *processor sharing* policy, each task is allocated an equal amount of processing units. Thus, processor sharing eliminates the slow truck effect: small jobs that arrive after large jobs will also get serviced right away. Therefore, the average delay under process sharing does not depend on the service time distribution but only on the mean service time and the arrival rate of the system. Processor sharing, FIFO and LIFO are all work conserving in that the system is idle if and only if there is no outstanding job in the system.

Several years ago, the notion of *generalized processor sharing* (GPS) or *fair queuing* (FQ) (see Section 2.1.9) was introduced and its implications and performance extensively investigated. Under GPS, the processing time for each outstanding job in the system is allocated not equally but according to a pre-determined fair share. GPS provides a flexible mechanism to treat jobs differently during busy or congested periods. Of course, processor sharing is just a simple case of GPS in which each outstanding task receives an equal share the system's capacity. Another dimension to the scheduling problem is that jobs can belong to different classes with different service time distributions and arrival rates.

The problem of scheduling packets of various traffic flows with different priorities in virtual circuit, connection-based wired packet networks has been modeled as a GPS problem [79, 78]. Although delay results related to process sharing are well known [57], the average delay for each class of node in GPS systems (that allows for arbitrary weights) is not easy to analyze. The authors in [79, 78] prove the delay

bounds of both single-server and multi-server GPS systems for networks, in which the maximum amount of data each source can transmit within any given interval is limited by a set of parameters.

There has not been much work in approximating GPS systems either. The authors of [14] develop a numerical approximation of the performance of a $M/M/1/\infty$ queue with GPS scheduling. They use hyper-exponential queues ($M/H_i/1$) to approximate average delay. However, the results are complex, require a lot of computation, and do not seem to lead to intuitive understanding.

As far as we know, there is no known general closed-form results for average delay, not conditional to service time, even for a single-server GPS system such as the one we are describing. There does exist an elegant closed-form solution when there are only two competing sessions and the service time distribution is exponential [30].

We do not present any new analytical delay results. However, we present a system description of wireless communication sessions within a GPS setting. We also derive worst-case relative ratios of session delay when there are only two entities competing for channel access.

Equilibrium Analysis

Little work has been done within a game theoretic framework to analyze the competition of channel access by rational nodes. Mangold conducts an extensive analysis on the application of game models to support QoS in situations in which two nearby 802.11e-based WLANs share a common channel [65]. It has been shown that the QoS mechanism specified in the 802.11e draft specification [44] is not effective in many cases especially when multiple APs share a common channel [66, 65]. In particular, the authors points out that the polling scheme specified as part of the centralized coordination scheme in the draft specification may not be able to poll client stations on time. Mangold shows that each node may benefit from a dynamic interaction, by adapting its behavior to the environment and the behaviors of other nodes [65]. Mangold also proposes strategies to be used by each 802.11e node so that the QoS for each node can be supported in the presence of competing 802.11e-based WLANs.

Unlike previous work, we develop a simple game theoretic model solely to examine the impact of fairness notions on achieving efficient equilibria under DCF and EDCF. The model and results are presented in [93, 95]. The context and goals of our analysis are very different from those of [65]. Using this model, we show how frame-based and bit-based fairness notions can lead rational nodes to use inefficient strategies to improve their own throughputs at the expense of degrading the aggregate throughputs. We demonstrate through analysis and simulation that a time-based fairness notion with a reasonable compensation model can force rational nodes to arrive at efficient equilibria while providing flexibility in scheduling transmissions to further improve aggregate performance.

2.2.7 Other Related Work

In AP-based WLANs, there is often an asymmetry in channel contention. The AP needs to access the channel to transmit data to multiple clients (*downstream* transmissions) whereas each client transmits data only to its associated AP (*upstream* transmission). Thus, the AP may need to contend for the channel more than each client. Bharghavan *et al.* proposes per-stream fairness as an alternative to per-node fairness [9]. The authors suggest that the number of backoff instance running at each node should be proportional to the number of flows. Our definition of per-link fairness is essentially the same as per-stream fairness. We use a similar approach to provide per-link fairness in AP-based WLANs.

Implementing any distributed MAC-layer protocol requires changes to proprietary firmware or hardware, where DCF is typically implemented. Thus, the implementation of a more robust and efficient solution (than DCF) is not a viable option for administrators who want to realize the benefits of time-based fairness in DCF-based networks.

In [94], we propose TBR, a link-layer scheduler that achieves a time-based, per-link fairness notion. TBR runs at the AP, works in conjunction with DCF, and requires no modifications to clients nor to DCF. We implemented TBR in the HostAP [50] driver and evaluated it on an 802.11b/Linux testbed. Based on a series of experiments reflecting realistic scenarios, we find that TBR is effective in allocating channel time equally among clients in the long-term. However, since TBR requires coordination between the AP and clients within each cell, TBR is not effective when nearby non-cooperative nodes fall under different administrative domains. We discuss TBR in detail in Chapter 6.

Under some conditions, some radios can correctly lock onto a stronger signal and receive a frame transmission despite interference from other frame transmissions. This effect is known as the *capture effect* [85] and it has been modeled and studied for many wireless networks including 802.11 networks [49, 85, 105]. The capture effect improves network efficiency since collisions do not necessarily lead to wasted capacity. However, under feedback based MAC protocols like 802.11, the capture effect can lead to a few nodes grabbing larger shares of channel capacity because of their successful transmissions during collisions, resulting in drastically unfair allocations of channel capacity. Unlike DCF and many existing MAC protocols, TES can provide fair allocations of capacity in the presence of capture effects.

Balachandran *et al.* propose that APs perform admission control and that users choose the access point with the least amount of load upon entering the network [6]. In their scheme, each mobile node explicitly expresses its lower and upper bounds of desired bandwidth. The admission control decision is made by a centralized admission control server that maintains the load information of all the APs, chooses the AP with the most available capacity, and suggests that nodes associate with it. Their work focuses on the system architecture for load balancing and admission control.

In [92], we address various challenges that need to be addressed to maximize the overall system throughput while maintaining high degree of fairness especially in the presence of communications among wireless clients in the same WLANs. The

algorithm described in [92] determines how best to associate each client with an AP and how much channel capacity to allocate to each flow. Schemes described in both [6] and [92] rely on each AP being able to have the perfect knowledge of the load of each client. In the absence of such information, however, TES's accurate estimation of the degree of contention in each cell could potentially help the AP in estimating the overall load and average load at each client.

Chapter 3

Analytical Model and Results

In this section, we identify and define various fairness notions, develop an analytical framework to understand the impact of each fairness notion on throughput and delay, and present results.

Specifically, we:

1. Develop a framework to compute the achieved throughput of each competing entity under different notions of fairness including time-based fairness and frame-based fairness,
2. Present a system description of wireless communication sessions within a GPS (Generalized Process Sharing) system model, through which average session wait time is derived for various fairness notions, and
3. Develop a game theoretic model to examine the impact of fairness notions on achieving efficient equilibria. Using that model, we prove that
 - DCF can lead rational nodes to inefficient equilibria, and
 - A MAC protocol that provides long-term time share guarantees can lead rational nodes to more efficient equilibria than DCF.

The next section examines various fairness notions. Sections 3.2-3.4 describe throughput analysis, wait time analysis and equilibrium analysis respectively. Chapter 4 compares the advantage and disadvantages of various fairness notions using throughput and delay results described in this chapter.

3.1 Fairness Notions for WLANs

In this section, we re-examine, in the context of wireless communication systems, the age-old problem of fair resource allocation among competing entities with equal priorities. We assume that all competing entities have equal priorities and defer our discussion on entities with different priorities until Section 3.2.3.

A resource allocation is *fair* if each *entity* that shares the *resource* receives an equal number of *fairness units* over a time interval, in which each entity is attempting to

use the resource. Although the resource is typically not open to interpretation, one fairness notion differs from another in its specifications of entities and fairness units. Fairness units include transmitted bits, transmitted frames, received bits, received frames, data transmission time units, etc.

In wireline systems, the capacity of the shared channel is fixed. So, a transmission schedule, produced by a fair queuing scheme, that allows each backlogged flow to transmit the same amount of bits always leads to equal divisions of the available channel capacity among competing flows, i.e., transmitted bits and transmission time units are equivalent. An alternation notion of fairness units could have been successfully received bits, which correspond closely to the utility of many applications. However, because of very low bit error rates on wired links (10^{-13}), each transmitted bit is almost always received successfully. Therefore, in the context of wire-line fair queuing, even under various definitions, achieving one fairness unit generally corresponds to using one resource unit.

In contrast, in multi-rate wireless communication systems, various notions of fairness units lead to different allocations of resource units among competing entities for the following reasons:

- The maximum achievable aggregate throughput of the shared channel varies depending on a particular allocation of transmission time among competing nodes transmitting at different data rates,
- Frame loss rates are relatively high because the wireless channel bit error rate (e.g. 10^{-5}) is much higher than in wired networks; loss rates are even higher because of collisions when nodes compete for channel access in a non-deterministic fashion (especially in CSMA systems), and
- Competing nodes often experience different loss rates because of location dependent channel errors.

Different fairness notions in WLANs often lead to different allocations of the underlying resource, resulting in different aggregate throughputs. The next three subsections discuss the resource, the entities and the fairness units respectively.

3.1.1 Resource

Unlike fairness notions, the shared resource is not open to value judgment or interpretation. In all wireless communication systems, competing nearby users are sharing the bandwidth of a common channel for a certain amount of time.

In FDMA systems, each user is assigned a fraction of the available channel bandwidth and not allowed to use the rest of the channel bandwidth. Thus, the resource that is being shared in FDMA systems is frequencies.

In TDMA and CSMA systems, users share a common channel with all the available bandwidth. Overlapping frame transmissions lead to collisions, resulting in wasted throughput. Therefore, only one nearby node should transmit at a time. The resource that is shared among users is *channel occupancy time*, the time available to transfer data.

Many wireless communication systems use hybrid approaches. For example, Bluetooth [11] uses both an FDMA technique and an TDMA technique by allocating a separate channel for each pair of users. The pair shares a common channel on a time-division basis. The 802.11 technologies also divide the available channel bandwidth into several orthogonal channels, each of which occupies a fraction of the available channel bandwidth. For example, there are 3 orthogonal channels for 802.11b and 802.11g devices to share. However, unlike true frequency-division systems like Bluetooth or cellular telephony systems, the 802.11 devices are not restricted to using only a single channel. I.e., each 802.11b or 802.11g device can exchange data in all three channels simultaneously. In practice, most 802.11 devices only communicate in a single channel at a time. This limitation is because of their simple single radio implementations. Recently, some vendors are beginning to produce more complex radio implementations that allow a device to transmit data across all three channels [27].

Our work focuses on issues only related to multiple users sharing a common channel by multiplexing channel occupancy time. Therefore, it applies to both TDMA and CSMA systems alike but does not apply to FDMA or CDMA systems.

3.1.2 Entities

An entity that shares the resource can be a link, a duplink or a node. A *link* is denoted by a pair of nodes, the sender and the receiver; frames are transmitted only from the sender to the receiver. A *duplink* (for duplex-link) is denoted by a pair of nodes that exchange frames in either direction. A *node* is denoted by an AP, a relay node or a client node that transmits frames to other nodes.

In AP-based cells, under per-link fairness, each client node will have an upstream link (to the AP) and/or a downstream link (from the AP). The AP can be associated with at most $2N$ links, where N is the number of client nodes. Each link is entitled to get the same number of fairness units. Thus, each client node in the same cell will get the same number of fairness units to transmit to the associated AP and the AP will also get that same amount to transmit to each client. So, if a client node transmits data to the AP as well as receives data from the AP, the total number of fairness units allocated to that client node for communications (in both upstream and downstream directions) can be twice as much as the number of fairness units allotted to a client that only communicates in one direction. Notice that, under per-link fairness, the AP is entitled to a larger share of the fairness units than each client node.

Under per-duplink fairness, a node with multiple duplinks can obtain a larger share of fairness units than a node with a smaller number of duplinks. In AP-based cells, such a fairness notion allows the AP to obtain fairness units proportional to the number of the AP's downstream clients. Thus, each backlogged client is entitled to an equal share of the shared fairness units to exchange frames with the AP in both upstream and downstream directions. Thus, per-duplink fairness can also be called *per-client fairness* in AP-based WLANs.

Under per-node fairness, each node will get the same number of fairness units to transmit data (to any other nodes). In AP-based cells, the AP will get the same number of fairness units to transmit (to all clients) as each client node gets to transmit

to the AP, leading to biases against clients that mainly have downstream traffic.

3.1.3 Fairness Units

The main fairness goal of a MAC layer is to allocate equal shares of *fairness units* among competing entities. Fairness units can be transmitted frames (*frame-based*), transmitted bits (*bit-based*), received bits (*received-bit-based*) or channel occupancy time for transmissions (*time-based*). Although these notions of fairness units are roughly equivalent in the context of fair queuing in wired networks, they differ significantly in the context of WLANs in how they allocate the shared resource (channel occupancy time) among competing nodes

Frame-based fairness and bit-based fairness are only equivalent when all entities use the same frame size. Transmitted bits and received bits are equivalent only if competing entities observe zero loss rates. Under time-based fairness, fairness units are equivalent to the underlying resource units. Under typical channel conditions and in the presence of rate diversity, each of these notions lead to different allocations of channel occupancy time.

For the rest of this chapter, when we say bit-based fairness, we mean *MAC-layer payload bits*, that exclude the MAC layer header and physical layer header. Under time-based fairness, the channel occupancy time allocated for a frame transmission is the total amount of time needed to transfer a MAC layer frame, including the amount of time necessary to transmit the MAC layer header as well as physical layer header and preamble.

3.2 Throughput Analysis

In this section, we examine the impact of the choice of fairness units on achieved throughput. Most of the results in this section appear in [94]. We validate our analysis through simulation in Section 4.3.

Let:

I : be the set of competing entities contending for channel access. Under per-node fairness, I represents the set of competing nodes, and under per-link fairness, I represents the set of competing transmitter and receiver pairs.

τ : be the duration, in seconds, during which all competing entities continuously wish to send data.

For each competing entity $i \in I$, we define the following terms that characterize its communication process during τ :

r_i : the data transmission rate (in bps) used by i ; d^{max} and d^{min} are the maximum and minimum data rates allowed respectively,

s_i : the MAC-layer per-frame payload size (in bits) used by i ; s^{max} is the maximum frame size allowed,

t_i : the channel occupancy time (in seconds) allocated to i during τ . The *channel occupancy time* necessary to transmit each data frame includes i) the transmission time of the data frame, ii) the transmission time of a synchronous MAC-layer *ack*, which is transmitted by the receiver some microseconds (10 under DCF) after successfully receiving the data frame and iii) the propagation delays. Each transmitted frame adds to the channel occupancy time used, irrespective of the failure or success of the transmission,

$g_i = (r_i, s_i)$: the transmission strategy, the pair (r_i, s_i) , used by i , and

$\alpha_i(g_i, I)$: the overall success rate observed as a fraction of the total number of transmitted bits. It is a function of the data rate and frame size used, and the level of contention and channel conditions experienced. Generally, losses caused by channel errors decrease with reduction in frame size or data rate. Losses caused by collisions decrease with decreased level of contention.

For simplicity, we assume that r_i and s_i are fixed for the duration of τ . However, the analysis we show in this section can be extended to cases where the data rate and frame size of an entity vary during τ .

In WLANs, an important time component is *idle time*. A typical CSMA MAC protocol requires competing entities to remain idle for randomized intervals to reduce collisions, as explained in Section 2.1.8. Therefore, each transmission event is typically preceded by an idle period. The idle time is not considered as a component of the channel occupancy time. For many CSMA protocols, the average amount of idle time preceding transmissions mainly depend on i) the number of entities and ii) their frame loss rates. Let

G_I : be the vector of transmission strategies used by entities during τ , i.e., $g_i \in G_I$ is the transmission strategy used by node i ,

α_I : be the vector of the success rates of all competing entities,

$f^{chan}(G_I, \alpha_I)$: be the fraction of channel occupancy time relative to τ when each entity i uses transmission strategy $g_i \in G_I$ and experiences success rate $\alpha_i \in \alpha_I$. $f^{chan}(G_I, \alpha_I)$ is MAC-specific since the idle time overhead varies with MAC protocol.

3.2.1 Assumptions

We make the following assumptions to simplify our analysis.

Assumption 1. *The channel is lossy but the error rate is bounded, i.e., $\alpha_i > 0$ for any interval τ .*

Assumption 2. *Frame losses on the wireless channel are caused only by channel errors and/or collisions.*

Assumption 3. *When multiple nearby nodes compete for channel access, at most one successful reception can be made among nodes that are within interference range.*

Assumption 4. $\alpha_I^{F1}(G_I) = \alpha_I^{F2}(G_I)$ for any two fairness notions $F1$ and $F2$.

Under the same set of transmission strategies and channel conditions, loss rates of entities attributable to channel errors will remain unchanged for different notions of fairness. In practice, depending on how transmission opportunities are allocated under each fairness notion, collision rates of entities may vary under different fairness notions. However, for simplicity, we assume that is not the case. Our assumptions mask many nuances of MAC and physical layers, but accurately capture the major impact of fairness notions on network performance.

Assumption 5. *Each entity always has data to transmit, i.e., is continuously backlogged.*

This is a common assumption, usually referred to as a *fluid traffic model*, in analyzing throughput [25, 102].

Assumption 6. *Each entity transmits only one frame in each transmission opportunity.*

This is the standard practice of most existing MAC protocols. However, there are proposals to allow multiple (back-to-back) frame transmissions in each transmission opportunity. We discuss on this issue in Section 3.4.

Assumption 7. *All nodes are assumed to be within radio range of each other.*

When this assumption is not true, it leads to the well-known *hidden terminal* problem, in which a node that cannot hear an existing transmission from another nearby node transmits, resulting in a collision at its receiver. However, the hidden terminal problem can be alleviated by a virtual carrier sensing mechanism such as the 802.11 specified RTS/CTS (request-to-transmit/clear-to-transmit) protocol that requires a sender to ask permission from its receiver before transmitting (see Section 2.1.8). Our analysis in this chapter can extend to MAC protocols with RTS/CTS enabled. In general, the assumption of nodes being within radio range does not affect the correctness of our analyses as shown in our simulation results throughout the thesis.

3.2.2 Achieved Throughput

Based on our definition of f_{chan} ,

$$\sum_{i \in I} t_i = f^{chan}(G_I, \alpha_I) * \tau \quad (3.1)$$

In practice, $f^{chan}(G_I, \alpha_I)$ does not equal to 1 and is MAC-protocol specific. Specifically, $f^{chan}(G_I, \alpha_I)$ depends on i) the amount of idle time preceding transmissions and ii) the amount of channel occupancy time attributable to collisions during τ . Idle time is not considered part of the channel occupancy time. When a collision occurs, the amount of channel occupancy time attributable to the collision is different from the elapsed time of the collision event. This is because the channel occupancy time

of the frame transmission of each node i involved in the collision is accounted for in t_i . For these reasons, $f^{chan}(G_I, \alpha_I) \neq 1$.

We now define $\gamma^{theo}(g_i)$, the theoretically achievable throughput as follows:

$$\gamma^{theo}(g_i) = \frac{s_i}{t_i^{ovh} + \frac{s_i + b_i^{ovh}}{r_i}} \quad (3.2)$$

where s_i is the payload size, b_i^{ovh} the bit overhead and t_i^{ovh} the time overhead per frame. The *bit overhead* represents the fixed number of bits that are required in each frame (but are not payload bits, such as the MAC layer header). Generally, b_i^{ovh} is the same for data frames of various sizes. The time overhead is the combined time necessary to transmit a physical layer preamble, physical layer-header, the synchronous acknowledgment and the interframe space time between the data and ack frames. We separate these two overhead components because the MAC layer header is transmitted at the same rate as the MAC layer payload, but the physical layer header and preamble are often transmitted at a pre-defined data rate that could be different from the one used to transmit MAC layer data.

For a given transmission strategy, $g_i = (r_i, s_i)$, $\gamma^{theo}(r_i, s_i)$ denotes the upper bound for the achieved throughput of entity i and can be computed for a given set of MAC layer parameters. Entity i 's achieved throughput will be less than $\gamma^{theo}(r_i, s_i)$ because of idle periods and frame losses.

We define $f_i(G_I)$ as the fraction of channel time achieved by entity i relative to the total amount of channel time achieved by all entities. That is

$$f_i(G_I) = \frac{t_i(G_I)}{\sum_{j \in I} t_j(G_I)} \quad (3.3)$$

We now derive the achieved steady-state MAC-layer throughput of a wireless entity $i \in I$ employing strategy $g_i = (r_i, s_i)$ when competing against one or more other entities during τ . The *achieved throughput* $\gamma_i(G_I)$ of entity i is:

$$\begin{aligned} \gamma_i(G_I) &= \alpha_i(g_i, I) * \gamma^{theo}(g_i) * \frac{t_i}{\tau} \\ &\text{Substituting Equation 3.1} \\ &= \alpha_i(g_i, I) * \gamma^{theo}(g_i) * \frac{t_i * f^{chan}(G_I, \alpha_I)}{\sum_{j \in I} t_j(G_I)} \\ &\text{Substituting Equation 3.3} \\ &= \alpha_i(g_i, I) * \gamma^{theo}(g_i) * f^{chan}(G_I, \alpha_I) * f_i(G_I) \end{aligned} \quad (3.4)$$

We define the practically achievable throughput, γ_i^{prac} , as the product of the first three terms:

$$\gamma_i^{prac}(G_I, \alpha_I, f^{chan}) = \alpha_i(g_i, I) * \gamma^{theo}(g_i) * f^{chan}(G_I, \alpha_I) \quad (3.5)$$

γ_i^{prac} is what entity i could practically achieve if it were granted 100% of the channel occupancy time (allocated to all competing entities), provided that i) the fraction of

idle time relative to the elapsed time remained unchanged and ii) the success rate remained unchanged. For convenience, we will use the following short-form notations: $\alpha_i(g_i)$ for $\alpha_i(g_i, I)$ and $\gamma_i^{prac}(g_i)$ for $\gamma_i^{prac}(G_I, \alpha_I, f^{chan})$.

The aggregate network throughput is simply:

$$\gamma_I(G_I) = \sum_{i \in I} \gamma_i(G_I) \quad (3.6)$$

3.2.3 Representing Fairness Notions

The shares of fairness units of competing nodes, specified by a fairness notion, can be translated to a desired fair share vector or a *weight* vector ϕ , in which ϕ_i denotes the share of the underlying resource units (not fairness units) that each entity i should achieve. Under this model, the ratio of the number of resource units allocated to entity i during interval τ to that allocated to entity j should be $\frac{\phi_i}{\phi_j}$.

In the context of WLANs, each type of fairness notion that we have described in this chapter can be captured with a corresponding fair share vector describing the desired allocation of channel occupancy time. Again, the choice of fairness units varies with each fairness notion. Therefore, ϕ can be used to provide a mapping in terms of how each unit of fairness unit maps to units of channel occupancy time.

Under frame-based fairness, each entity i is allocated the same number of frame transmissions. Therefore, the fair share of channel occupancy time allocated to each entity i is simply the amount of channel time required by entity i to transfer one fairness unit (a frame).

$$\phi_i^{FF} = \frac{s_i}{\gamma_i^{theo}(g_i)} \quad (3.7)$$

Under bit-based fairness, each entity i gets to transmit the same number of payload bits, irrespective of its frame size. Thus, ϕ_i^{BF} is simply the amount of channel time required to transmit one payload bit.

$$\phi_i^{BF} = \frac{1}{\gamma_i^{theo}(g_i)} \quad (3.8)$$

ϕ_i^{BF} is inversely proportional to the per-frame payload size (see Equation 3.2). That is if entity i and entity j uses the same data rate but different payload sizes, say $s_i > s_j$, then $\phi_i^{BF} < \phi_j^{BF}$. More channel time needs to be given to entity i , which incurs more MAC and physical layer overheads per payload bit.

Under time-based fairness, each entity i achieves the same amount of channel occupancy time. The fairness units and the resource units are equivalent.

$$\phi_i^{TF} = 1 \quad (3.9)$$

ϕ_i^{TF} does not depend on g_i or anything else.

The fraction of channel occupancy time allocated to each entity i can also be

represented as a function of ϕ as:

$$f_i(G_I) = \frac{\phi_i}{\sum_{j \in I} \phi_j} \quad (3.10)$$

3.2.4 Achieved MAC-layer Throughput Under Various Fairness Notions

In this section, we describe the fraction of channel time allocated to entity i , the achieved throughput of entity i and the total achieved throughput under each fairness notion, using equations described earlier.

We can re-state Equations 3.4, 3.6 and 3.10 for each fairness notion. Under frame-based fairness, we have:

$$f_i^{FF} = \frac{\frac{s_i}{\gamma^{theo}(g_i)}}{\sum_{j \in I} \frac{s_j}{\gamma^{theo}(g_j)}}; \gamma_i^{FF} = \frac{s_i * \alpha_i(g_i) * f^{chan}}{\sum_{j \in I} \frac{s_j}{\gamma^{theo}(g_j)}}; \gamma_I^{FF} = \frac{\sum_{j \in I} s_j * \alpha_j * f^{chan}}{\sum_{j \in I} \frac{s_j}{\gamma^{theo}(g_j)}} \quad (3.11)$$

Under bit-based fairness, we have:

$$f_i^{BF} = \frac{\frac{1}{\gamma^{theo}(g_i)}}{\sum_{j \in I} \frac{1}{\gamma^{theo}(g_j)}}; \gamma_i^{BF} = \frac{\alpha_i(g_i) * f^{chan}}{\sum_{j \in I} \frac{1}{\gamma^{theo}(g_j)}}; \gamma_I^{BF} = \frac{\sum_{j \in I} \alpha_j * f^{chan}}{\sum_{j \in I} \frac{1}{\gamma^{theo}(g_j)}} \quad (3.12)$$

And, under time-based fairness, we have:

$$f_i^{TF} = \frac{1}{|I|}; \gamma_i^{TF} = \frac{\alpha_i(g_i) * \gamma^{theo}(g_i) * f^{chan}}{|I|}; \gamma_I^{TF} = \frac{\sum_{j \in I} \alpha_j * \gamma^{theo}(g_j) * f^{chan}}{|I|} \quad (3.13)$$

Observe that, under FF and BF, the achieved throughput of each entity i is dependent on the theoretically achievable throughputs of all entities (i.e., the denominator), which in turn depends on their data rates and frame sizes. However, the achieved throughput of each entity i under TF depends only on its practically achievable throughput not on the transmission strategies of other entities.

That is only TF has the following property:

Independence Property The long-term throughput of an entity (node) competing against any number of entities (nodes) running at different speeds is equal to the throughput that the node would achieve in an existing single-rate 802.11 WLAN in which all competing entities (nodes) were running at its rate.

We validate our throughput analysis through simulation and provide a comparison among various fairness notions in the next chapter.

3.3 Delay Analysis

In this section, we present a system description of wireless communication sessions within an established GPS (Generalized Process Sharing) system model [58]. Much previous work analyzed GPS systems in the context of sharing CPU cycles among jobs [14, 30, 58] and in the context of leaky bucket constrained wired networks [79, 78]. We do not present new delay results but show how to apply known results that are relevant within the context of our system description.

We also derive worst-case relative ratios of session delay when there are only two entities competing for channel access. The next section describes how we model communication sessions with a GPS framework. The following section describes our derivations of worst-case relative ratios of session delay.

3.3.1 Average Wait Time under Various Fairness Notions

We model a single-AP multi-rate WLAN with one or more competing communication sessions as a GPS system serving one or more tasks. A communication session of an entity requiring wireless channel time can be considered as a task requiring CPU cycles of the GPS system. In a traditional GPS system, the processor capacity is allocated among multiple tasks according to their fair shares or weights. Similarly, competing communication sessions will achieve the amounts of channel occupancy time according to their fair shares or weight vector.

Each communication session is associated with an entity and there is at most one session associated with each entity. The entity transmits and/or receives multiple frames during each session, depending on the definition of entities. If entities are nodes, in each session, its associated entity (either the AP or a client) transmits data. If entities are links, in each session, the source node of its associated link transfers data to the destination node. If entities are duplinks, in each session, the end nodes of its associated duplink exchange data.

Let:

O : be the set of communication sessions,

C : be the set of ordered priority classes to which communication sessions belong,

l_i : be the length or the amount of data to be transferred in session i in bits; session i ends when l_i has been transferred, and

c_i : be the priority class that session i belongs to,

d_i : be the *wait time* or delay of session i , i.e., the duration between the time the first data bit is transmitted and the time the last data bit is transmitted,

x_i : be the *service time* or the minimum amount of time required to complete session i if it were given 100% of the channel time,

n_c : be the number of class c sessions, and

μ_c : be the mean service time of class c sessions

We will also use the notations defined in earlier sections: ϕ_i for the fair share of channel occupancy time of session i , $g_i = (r_i, s_i)$ for the transmission strategy of session i , and α_i for the success rate of session i .

Our goal is to understand the impact of fairness notions on the average wait time, $E[d]$. Different notions of fairness units lead to different fair shares of channel time allocation (ϕ) among competing entities, leading to different average session wait time.

For each session i during a congested interval, the minimum amount of time required to transfer l_i can be computed as a function of γ_i^{prac} , the practically achievable throughput of session i if it were granted 100% of the channel occupancy time. Thus, we have:

$$x_i = \frac{l_i}{\gamma_i^{prac}} = \frac{l_i}{\alpha_i * \gamma^{theo}(g_i) * f^{chan}(G_I, \alpha_I)} \quad (3.14)$$

Note that $d_i \leq x_i$.

Many communication sessions will have similar characteristics of g_i and α_i . For instance, two nearby 802.11b devices that are both close to the AP may experience similar low loss rates and use the maximum data rate of 11 Mbps. Therefore, we consider sessions with similar characteristics as one class, whose arrival process and service time distributions are modeled collectively. For a class, c , we can describe its session characteristics using $E[x_c]$, the expected service time of class c sessions, and ϕ_c the (average) fair share of channel occupancy time of class c sessions. As shown in Equation 3.14, $E[x_c]$ can be computed using α_c and g_c , representing the (average) success rate and transmission strategy of class c sessions. The departure rate (or session completion rate), μ_c , of class c sessions is:

$$\mu_c = \frac{1}{E[x_c]} \quad (3.15)$$

Each class $c \in C$ session among the set of competing sessions, O , is guaranteed to achieve the following fraction of channel occupancy time:

$$f_c = \frac{\phi_c}{\sum_{k \in C} \phi_k * n_k} \quad (3.16)$$

where n_k is the number of competing sessions belonging to class k . Note that $|O| = \sum_{c \in C} n_k$. A class c session is being serviced (in terms of channel occupancy time units) at a rate no smaller than f_c .

We consider a GPS system with a Poisson arrival process at each session class c with the mean arrival rate λ_c , and an exponential service time distribution with the mean service time of $\frac{1}{\mu_c}$. We are not aware of closed-form solutions to compute the expected wait time $E[d_c]$ of class c sessions under this system for a $C > 2$.

Let there be two classes of sessions: i and j . Let A be a fairness notion, under

which the fair share of channel occupancy time of class i and class j are $\phi_i > 0$ and $\phi_j > 0$ respectively. The expected wait time of class i sessions and that of class j sessions under the exponential service time distribution with means μ_i and μ_j and the Poisson arrival processes with rates λ_i and λ_j are as follows [30]:

$$E[d_i^A] = \frac{1}{\mu_i(1-\rho)} \left[1 + \frac{\mu_i \rho_j (\phi_j - \phi_i)}{\mu_i \phi_i (1 - \rho_i) + \mu_j \phi_j (1 - \rho_j)} \right] \quad (3.17)$$

$$E[d_j^A] = \frac{1}{\mu_j(1-\rho)} \left[1 + \frac{\mu_j \rho_i (\phi_i - \phi_j)}{\mu_i \phi_i (1 - \rho_i) + \mu_j \phi_j (1 - \rho_j)} \right], \quad (3.18)$$

where $\rho_i = \frac{\lambda_i}{\mu_i}$ and $\rho = \rho_i + \rho_j$. ρ is the total channel utilization. ρ must be less than 1 for the system to be stable, i.e., so that the expected delay does not go to ∞ .

As shown in the equations, the expected wait time of class i sessions depends on its fair share of channel occupancy time, ϕ_i , and that of the competing class. In Section 3.2.3, we showed ϕ_c for various notions of fairness, including time-based fairness and bit-based fairness. Using these results, we will compare the expected wait time under various fairness notions in the next chapter.

3.3.2 Worst-case Relative Ratio of Wait Time in a Two-Entity Competition

In this subsection, we analyze the impact of different time share allocations on user wait time in terms of worst-case and best-case schedules (produced under one fairness notion relative to another) when the system only has two sessions, i and j belonging to two different classes. For convenience, we will use the same identifiers i and j to denote the corresponding classes as well. Assume that $r_i > r_j$ and that $\gamma_i^{prac}(g_i) > \gamma_j^{prac}(g_j)$. Furthermore, assume that both sessions are started at the same time, $t = 0$. Therefore, the wait time or delay of each session is the same as its completion time.

Our goal is to understand for any two fairness notions, what the worst-case relative ratios of individual and aggregate wait times for any L , the vector of session size. The simple analysis described in this section is intended to provide additional insights that complement the findings in the previous section.

Worst-case Relative Wait Time Ratio

The worst-case (maximum) ratio of the wait time of a session under the schedule produced by a GPS server under policy A and that of the same session under policy B for any mix of session sizes (L) is:

$$WorstIndivRatio(A, B) = \max_L \left\{ \frac{d_i^A}{d_i^B} \right\} \quad (3.19)$$

We first examine the completion time of each session under a particular fair allocation. Let ϕ be the fair share vector. Depending on l_i and l_j , either session i or session j can complete first. When session i completes first, the wait time of each

session is as follows:

$$d_{i|i=first} = \frac{l_i}{\phi_i * \gamma_i^{prac}(g_i)} \quad (3.20)$$

$$d_{j|i=first} = d_{final} \quad (3.21)$$

where

$$\begin{aligned} d_{final} &= x_i + x_j \\ &= \frac{l_i}{\gamma_i^{prac}} + \frac{l_j}{\gamma_j^{prac}(g_j)} \end{aligned} \quad (3.22)$$

We use $i = first$ to denote that session i completes first. d_{final} is the time the last session is completed and is the same under any policy so long as the system is work conserving. In other words, the total amount of time required to transfer all data (or the time the last session completes) under any work-conserving policy is the same. However, both the individual wait time for the first session to complete and average wait time vary from policy to policy.

Let Fi and Fj be two different fairness notions. Under policy Fi (for favor i), session i is allocated a higher time share than that allocated under policy Fj , i.e., $\phi_i^{Fi} > \phi_i^{Fj}$. It follows that $\phi_j^{Fi} < \phi_j^{Fj}$. Again, under both policies the completion time of the last session is the same, since both policies are work conserving.

Clearly, under any fairness notion A , $d_i^A \geq x_i$. Under SRPT, a session with the smallest d^{req} will be always given 100% of the time, minimizing the average wait time. However, in our model, l_i and l_j are unknown. Therefore, neither Fi nor Fj will be able to minimizing the average wait time for any pair of l_i and l_j . x_i and x_j along with ϕ^{Fi} and ϕ^{Fj} will determine the wait time of i and j .

There are four scenarios under which the aggregate wait time under Fi is different from that under Fj , depending on which session completes first under each fairness notion.

First, session i may complete earlier than session j under both policies. In this case, the wait time of session i under Fi will be smaller than that under Fj , since $\phi_i^{Fi} > \phi_i^{Fj}$, whereas the wait time of session j is the same under both policies. We label this situation Case-1.

Second, session i may complete earlier under Fi but may not so under Fj . There are two sub-cases. In the first sub-case (called Case-2a), the wait time of session i under Fi ($d_{i|i=first}^{Fi}$) is smaller than the wait time of session j under Fj ($d_{j|j=first}^{Fj}$). In the second sub-case (called Case-2b), $d_{i|i=first}^{Fi} > d_{j|j=first}^{Fj}$.

Third, session j may complete earlier than session i under both policies. In this case (called Case-3), it is clear that the wait time of session j under Fi is larger than that under Fj since $\phi_j^{Fi} < \phi_j^{Fj}$.

Finally, session j may complete earlier than session i under Fi but may not under Fj . However, this scenario is not possible. Since, $\phi_j^{Fi} < \phi_j^{Fj}$, if session j completes first under Fi , it must be that session j also completes first (not last) under Fj .

Using Equations 3.20 and 3.21, we derive the necessary relationship between x_i and x_j for each of the four cases mentioned, Case-1, Case-2a, Case-2b and Case-3.

Case	Wait Time Relationship	Necessary Condition
1	$d_{i i=first}^{Fi} < d_{i i=first}^{Fj}$	$\frac{x_i}{x_j} < \frac{\phi_i^{Fj}}{\phi_j^{Fj}}$
2a	$d_{i i=first}^{Fi} < d_{j j=first}^{Fj}$	$\frac{\phi_i^{Fj}}{\phi_j^{Fj}} \leq \frac{x_i}{x_j} < \frac{\phi_i^{Fi}}{\phi_j^{Fi}}$
2b	$d_{i i=first}^{Fi} \geq d_{j j=first}^{Fj}$	$\frac{\phi_i^{Fi}}{\phi_j^{Fi}} < \frac{x_i}{x_j} \leq \frac{\phi_i^{Fj}}{\phi_j^{Fj}}$
3	$d_{j j=first}^{Fj} > d_{j j=first}^{Fi}$	$\frac{\phi_i^{Fi}}{\phi_j^{Fi}} < \frac{x_i}{x_j}$

Table 3.1: Comparison of the wait time of a session that completes no later than another session under Fi and that under Fj .

Table 3.1 describes each case and the necessary condition for that case.

Under Case-1, the wait time of any session under Fi is no greater than or equal to that of the same session under Fj . Therefore, Case-1 is not of interest to us in finding the worst-case relative wait time ratio.

Under Case-2a and Case-2b, $d_{i|i=first}^{Fi} \geq d_{i|j=first}^{Fj}$. Thus, worst-case relative ratio of session wait time under Fi to that under Fj is:

$$\begin{aligned}
WorstIndivRatio^{2a,2b}(Fi, Fj) &= \max_L \left\{ \frac{d_{j|j=first}^{Fi}}{d_{j|j=first}^{Fj}} \right\} \\
&= \max_L \left\{ \frac{\frac{x_i + x_j}{x_j}}{\frac{\phi_j^{Fj}}{\phi_j^{Fi}}} \right\} \\
&= \max_L \left\{ \frac{x_i}{x_j} * \phi_j^{Fj} + \phi_j^{Fj} \right\} \\
&= \max_L \left\{ (\frac{x_i}{x_j} + 1) * \phi_j^{Fj} \right\}
\end{aligned}$$

Thus, $WorstIndivRatio(Fi, Fj)$ is maximized when $\frac{x_i}{x_j}$ is maximized. However, this ratio cannot be arbitrarily large. Under Case-2a and Case-2b, $\frac{x_i}{x_j} \leq \frac{\phi_i^{Fi}}{\phi_j^{Fi}}$. Therefore,

$$\begin{aligned}
WorstIndivRatio^{2a,2b}(Fi, Fj) &= (\frac{\phi_i^{Fi}}{\phi_j^{Fi}} + 1) * \phi_j^{Fj} \\
&= \frac{\phi_i^{Fi} + \phi_j^{Fi}}{\phi_j^{Fi}} * \phi_j^{Fj} \\
&= \frac{\phi_j^{Fj}}{\phi_j^{Fi}}
\end{aligned}$$

Under Case-3, $d_{i|j=first}^{Fi} = d_{i|j=first}^{Fj}$. Thus,

$$\begin{aligned}
WorstIndivRatio^3(Fi, Fj) &= \max_L \left\{ \frac{d_{j|j=first}^{Fi}}{d_{j|j=first}^{Fj}} \right\} \\
&= \max_L \left\{ \frac{\frac{x_j}{\phi_j^{Fi}}}{\frac{x_j}{\phi_j^{Fj}}} \right\} \\
&= \max_L \left\{ \frac{\phi_j^{Fj}}{\phi_j^{Fi}} \right\} \\
&= \frac{\phi_j^{Fj}}{\phi_j^{Fi}}
\end{aligned}$$

Thus, for all three cases,

$$WorstIndivRatio(Fi, Fj) = \frac{\phi_j^{Fj}}{\phi_j^{Fi}} \quad (3.23)$$

Observe that $WorstIndivRatio(Fi, Fj)$ only depend on fair shares under Fi and Fj and does not depend on l_i, l_j, g_i, g_j , etc.

The analysis presented in this and the previous sections allow us to evaluate the impact of fairness notions on achieved throughput and session wait time. Using the results described in these sections, we quantify the advantages and disadvantages that one fairness notion has over another in the next chapter.

3.4 Equilibrium Analysis

In this section, we develop a game theoretic model to examine the impact of fairness notions on achieving efficient equilibria by rational players in non-cooperative environments. In *non-cooperative environments*, nodes (*players*) will attempt to maximize their own utility without cooperating with other players. For example, neighboring wireless networks that share a common channel are often managed by different administrative authorities (e.g. WLANs operated by neighboring small enterprises in a multi-floor commercial building) and thus centralized coordination and allocation of resources in these systems is impractical. This is a major reason why 802.11-based WLANs run a distributed MAC protocol.

In many non-cooperative environments, nodes may attempt to choose a transmission strategy in a rational manner. A *rational* node: i) evaluates its potential utilities when competing against other rational nodes, ii) chooses a transmission strategy that yields the highest utility given the transmission strategies used by other nodes.

Each rational node can strategically choose i) the data rate and ii) the frame size of each frame transmission to maximize its utility. 802.11 specification allows flexible use of these two MAC-layer parameters. Most manufacturers of 802.11 wireless interface cards include proprietary data rate protocols to adaptively select the data transmission rate based on observed channel conditions, with the goal of reducing frame loss rates and/or maximizing aggregate throughput [22, 26, 75, 23], but not

necessarily on maximizing the achieved throughput of an individual node. However, a user can disable these protocols and implement its own rational, non-cooperative rate adaptation scheme. In principle, a manufacturer may also have an incentive to implement a rational non-cooperative rate adaptation scheme so that its device gains higher achieved throughput when competing against devices from other manufacturers.

Unfortunately, as we demonstrate in this thesis using our game theoretical model and simulation results, the popular 802.11 MAC protocol DCF and its enhanced version EDCF lead to significantly degraded performance in the presence of rational, non-cooperative competition for channel access.

For simplicity, in this section we conduct our analysis on UDP flows. We note that there is a one-to-one correspondence between a UDP packet and a MAC layer frame. TCP complicates the analysis since we need to take into account frame loss rates in both directions, one for TCP data packets and the other for TCP ack packets. However, our simulation results in Section 3.4.7 show that the end results of our theorems also hold for TCP flows. Most of the results in this section appears in [93, 95].

3.4.1 Game Model

In this section, we model two rational, non-cooperative nodes, i and j , each sending UDP data to a receiver as two players playing a finitely repeated non-cooperative game. In each stage, stagegame Gm is played as follows. The first node (say i) transmits a burst of $b_i \leq n_i$ frames successively. Following that, the second node transmits a burst of up-to n_j frames successively. This model is a more general model than the one we assumed in earlier sections in which each node can only transmit one frame in each transmission opportunity (see Assumption 6 in Section 3.2.4). This extension is necessary to model EDCF-like protocols. Recall that EDCF is an enhanced cousin of DCF as specified in the 802.11e draft specification [44].

For simplicity, we still assume that the amount of idle time required by the MAC protocol as a fraction of the total amount of channel occupancy time remains unchanged under different fairness notions so long as the loss rates experienced remains unchanged. That is f^{chan} depends only on the number of nodes. Again, this assumption simplifies the analysis but does not undermine the results presented in this chapter.

Under our assumption that nodes are always backlogged, each node will attempt to transmit the maximum numbers of frames allowed. However, the actual number of frames transmitted (b_i) may be less than the maximum allowed depending on the backoff technique used by the MAC protocol.

A stagegame may last no more than τ seconds. At the beginning of each stagegame, with probability p node i communicates first and with probability $1 - p$ node j communicates first. For the rest of this chapter, we assume that $p = 0.5$ and consider a K -repeated game $Gm(K)$ in which the stagegame Gm is played K times and K is even. The values of n , m and τ are dictated by the underlying MAC protocol.

The utility of each player is its achieved UDP throughput over $\tau * K$ seconds.

At each stagegame, the available actions of each player, or the *strategy space*, are to set its data rate and to set its frame size. The goal of each competing player is to employ the strategy $g^* = (r^*, s^*)$ that maximizes its achieved throughput given the other player's best transmission strategy. Therefore, the strategy space for players is the set of possible data rate and frame size pairs. We do not consider other possible strategies such as transmit power.

3.4.2 Nash and Subgame Perfect Equilibriums

In each stagegame Gm , nodes are in a Nash Equilibrium (NE) if neither node has incentive to deviate from its current strategy of using a specific combination of data rate and frame size. There could be more than one NE in each stagegame.

An outcome of a K -repeated game $Gm(K)$ is the achieved throughputs of the two nodes given their strategies over all K stagegames. To simplify our analysis, we assume that the overall channel conditions remain relatively unchanged. In other words, for a given data rate and frame size, the success probability of frame transmission observed by a node over each interval of τ seconds in each stagegame is similar. While this assumption is typically valid for relatively static environments where channel errors occur randomly, it is not valid for mobile environments in which a moving sender or receiver can lead to correlated frame losses on a short timescale [71, 87, 83]. That is channel conditions of some stagegames may drastically differ from that of other stagegames in mobile environments. We will deal with this issue in Section 3.4.6.

A *subgame* beginning at stage $k + 1$ of $Gm(K)$ is the repeated game in which stagegame Gm is played $K - k$ times and is denoted $Gm(K - k)$. An outcome of a K -repeated game ($Gm(K)$) is considered *subgame perfect* if in each subgame, only NE strategies are played. In general, there could be many *subgame perfect equilibriums* (SPEs) for a K -repeated game since there could be more than one NE. However, if stagegame Gm has a unique NE, then the finitely repeated game $Gm(K)$ also has a unique SPE, in which the unique NE of Gm is played at every stagegame [35].

Ideally, each node should use a strategy that yields the maximum practically achievable throughput, leading to the maximum aggregate achieved throughput with respect to the particular channel time allocation. Therefore, an outcome in which each node employs a strategy yielding the maximum achievable throughput is considered *desirable*. A NE is considered *desirable* if its outcome is desirable and otherwise is considered *undesirable*. Similarly, a SPE of a K -repeated game is *desirable* if a desirable NE is played at each subgame and otherwise *undesirable*.

In non-cooperative environments, a rational player may use a strategy that yields non-optimal practically achievable throughput but achieves a higher time share ($Frac_i$), thereby, achieving higher throughput for that node. As a result, one or more undesirable NEs may exist in the stagegame. Nonetheless, when there exists at least one desirable NE in the stagegame, a desirable SPE (for the K -repeated game) can still be reached since rational nodes can use threats of retaliation to force a desirable SPE [35]. However, when the stagegame has a unique NE and that NE is undesirable, the resulting unique SPE of the K -repeated game is also undesirable.

The rest of this section shows that DCF, in many situations, and EDCF, in some

situations, lead rational nodes to arrive at undesirable unique NEs (and thus undesirable unique SPEs). Naturally, one might ask whether it is possible to design the MAC protocol so that it can always lead to desirable SPEs in non-cooperative environments. We show in Sections 3.4.5 and 3.4.7 that this indeed is possible.

3.4.3 Error Model

The analyses conducted in earlier sections are concerned only with long-term success rate. The long-term success rate of entity i using g_i is represented as a fraction of the total number of bits transmitted as $\alpha_i(g_i)$. We now look at the impact of two different patterns of loss.

For the rest of this chapter, we assume that each entity i uses the same frame size for each transmission. We begin with a simple error model under which each transmitted frame can be lost due to channel errors with a certain probability and independently of whether other transmitted frames have errors. In Section 3.4.6, we will extend this model to capture channel conditions that are time-correlated on short time scales and thus the lost probability of successive frames is not independent.

Furthermore, we assume that each collision event is independent of another. Therefore, when only one frame is transmitted in each burst, we can assume that frames of entity i are lost with a probability of $(1 - \alpha_i(g_i))$ due to channel errors and/or collisions and independently of whether other transmitted frames of i are lost. We note that when i transmits a number of successive frames during a transmission event and a collision happens, the lost probability of subsequent frame transmissions is not independent of that of the first frame involved in the collision. However, in our example, we are only focusing on two-node competitions and thus the typical loss rate due to collisions is relatively smaller compared to high frame loss rates due to channel errors. Therefore, for the first half of our discussion, we ignore this effect and assume that frame losses of entity i are independent of each other. We will consider an extended loss model under which this assumption is not true in Section 3.4.6.

3.4.4 Analysis of DCF

The achieved steady-state UDP throughput of a wireless node i employing strategy $g_i = (r_i, s_i)$ when competing against node j employing $g_j = (r_j, s_j)$ during interval τ is given in Equation 3.4 in Section 3.2 but restated here for convenience.

$$\gamma_i(G_I) = \alpha_i(g_i) * \gamma^{theo}(g_i) * f^{chan}(G_I, \alpha_I) * \frac{t_i}{\sum_{j \in I} t_j}$$

where t_i is the amount of channel occupancy time allocated to node i during τ .

DCF gives an equal long-term channel access probability to each contender with similar channel conditions [59, 97]. However, when two nodes experiencing different loss rates compete, the long-term channel access probability of the node with the higher loss rate will be lower. This is caused by the backoff algorithm that forces a node to backoff longer whenever it experiences a failed transmission. Our results

for DCF hold regardless of the existence of such a design. For simplicity, we ignore this effect. Thus, we assume that under DCF, competing nodes sending data frames over the same time interval will be able to transmit approximately equal numbers of frames irrespective of their frame loss rates. Furthermore, we assume that when nodes i and j are competing for channel access under DCF, every even frame transmission burst is transmitted by either node i or node j with 0.5 probability and every odd frame transmission burst is transmitted by a node that is different from the node which transmitted the previous transmission burst. Of course, in general, under DCF, frame transmissions on the short-term scale will not be as deterministic as our assumption. Nonetheless, with these simplifying assumptions, we have captured the long-term fair channel time allocation of DCF as well as a degree of randomness in the schedule. We call our DCF version *sDCF* (for simplified DCF).

Note that sDCF only allows a single frame to be transmitted during each transmission opportunity. Therefore, when nodes use sDCF, we can specify the game as follows: $b_i = b_j = n_i = n_j = 1$.

In the rest of this section, we prove theorems and claims using concrete examples and intuitions. More rigorous and formal arguments for these claims and theorems can be found in Appendix A.

Lemma 3.4.1. *Under sDCF, the amount of channel occupancy time each node achieves during each stagegame is the amount of time required to transmit its data frame using its transmission strategy. I.e., $t_i = \frac{s_i}{\gamma^{theo}(g_i)}$ and $t_j = \frac{s_j}{\gamma^{theo}(g_j)}$.*

Proof. Since each node only gets to transmit one frame each in a stagegame ($b_i = b_j = 1$), the claim is self-evident. \square

The achieved throughput of node i under sDCF (given frame-based fairness) is:

$$\begin{aligned}\gamma_i^{sDCF} &= \alpha_i(g_i) * f^{chan}(\alpha_I, G_I) * \gamma^{theo}(g_i) * \frac{\frac{s_i}{\gamma^{theo}(g_i)}}{\sum_{j \in I} \frac{s_j}{\gamma^{theo}(g_j)}} \\ &= \gamma_i^{prac}(g_i) * \frac{\frac{s_i}{\gamma^{theo}(g_i)}}{\sum_{j \in I} \frac{s_j}{\gamma^{theo}(g_j)}}\end{aligned}$$

Definition 3.4.2. Let G be the set of all available transmission strategies. Let g_i^* be the maximally efficient strategy that node i would use if it alone occupies the channel, i.e., $\forall g \in G, \gamma^{theo}(g_i^*) * \alpha_i(g_i^*) \geq \gamma^{theo}(g) * \alpha_i(g)$. Let g_i and g_j be the strategies of nodes i and j at a NE. g_i is *undesirable* if $\gamma_i^{prac}(g_i) > \gamma_i^{prac}(g_i^*)$. And thus, the NE is also undesirable.

Theorem 3.4.3. *Under certain channel conditions, there exist undesirable unique SPEs under sDCF.*

Proof by construction. Assume that there are two data rates r_1 and r_2 and that $r_1 > r_2$. Also, assume that each node uses maximum-sized frames and thus there are only two viable strategies that each node can choose: $g_1 = (r_1, s^{max})$ and $g_2 = (r_2, s^{max})$. $s^{max} = 1472$ bytes of UDP payload (the maximum MAC-layer frame size

of 1500 bytes minus 28 bytes for IP and UDP headers) and $f^{chan}(G_I, \alpha_I) = 0.75$ (the proof works with any value of $f^{chan}(G_I, \alpha_I)$.) Table 3.2 lists the theoretically achievable UDP throughputs under all four possible 802.11b data rates.

Strategy	d (Mbps)	s (bytes)	γ^{theo}	$\gamma^{theo} * f^{chan}$
g_{11}	11.0	1472	8.76	6.991
$g_{5.5}$	5.5	1472	4.711	3.863
g_2	2.0	1472	1.840	1.509
g_1	1.0	1472	0.915	0.750

Table 3.2: Theoretically achievable UDP throughputs under all possible 802.11b data rates.

Furthermore, assume that frame success rates of node i and j at different data rates are as shown in Table 3.3. According to the table, node j suffers little loss at any data rate. However, node i experiences a high loss rate when g_{11} is used and very low loss rates with the rest of the strategies. We focus on the strategies using the top two data rates (g_{11} , and $g_{5.5}$) since frame loss rates do not change much even when g_2 and g_1 are used.

x	$\alpha_x(g_{11})$	$\alpha_x(g_{5.5})$	$\alpha_x(g_2)$	$\alpha_x(g_1)$	$\gamma_x^{prac}(g_{11})$	$\gamma_x^{prac}(g_{5.5})$	$\gamma_x^{prac}(g_2)$	$\gamma_x^{prac}(g_1)$
i	0.63	0.99	0.99	0.99	4.404	3.824	1.494	0.743
j	0.99	0.99	0.99	0.99	6.921	3.824	1.494	0.743

Table 3.3: Frame success rates and corresponding practically achievable throughputs of node i and j at all possible transmission strategies.

Based on the equation stated in Section 3.4.4, we can compute pairs of the channel time fractions and the achieved throughputs of node i and node j for each possible combination of each node using g_{11} or $g_{5.5}$. This is shown in Table 3.4. Each pair in either the third or fourth row represents the fraction of channel occupancy time allocated to node i and that allocated to node j . Similarly, each pair in either the sixth or seventh row represents denotes the achieved throughputs of node i and node j . For instance, the pair in the third column of the sixth row, (1.56, 2.46), denotes that the achieved throughputs of nodes i and j are 1.56 and 2.46 Mbps respectively when node i uses g_{11} and node j uses $g_{5.5}$.

Based on the achieved throughput pairs, there exists a unique NE in which node i plays strategy $g_{5.5}$ and node j plays strategy g_{11} . However, this unique NE is undesirable. It is easy to see that if node i is the only one transmitting, the most efficient strategy is clearly g_{11} , i.e., $\gamma_i^{prac}(g_{11}) > \gamma_i^{prac}(g_{5.5})$. Similarly, g_{11} is the most efficient strategy for node j . Unfortunately, at the unique NE, node i uses a less efficient strategy $g_{5.5}$. As a result, the aggregate throughput at equilibrium ($2.36 + 2.46 = 4.82$) is less than the aggregate throughput of ($2.20 + 3.46 = 5.76$)

g_j	g_{11}	$g_{5.5}$
g_i	$(f_i(G_I), f_j(G_I))$	$(f_i(G_I), f_j(G_I))$
g_{11}	(0.50, 0.50)	(0.356, 0.644)
$g_{5.5}$	(0.644, 0.356)	(0.50, 0.50)
	$(\gamma_i(g_i), \gamma_j(g_j))$	$(\gamma_i(g_i), \gamma_j(g_j))$
g_{11}	(2.20, 3.46)	(1.56, 2.46)
$g_{5.5}$	(2.36, 2.46)	(1.91, 1.91)

Table 3.4: Fractions of channel occupancy time and achieved throughputs of node i and j when each node is using either g_{11} or $g_{5.5}$. The unique NE strategies for i and j are $g_{5.5}$ and g_{11} respectively.

when both nodes use their most efficient strategies and each is given 50% of the channel occupancy time.

Since the stagegame has a unique NE, the finitely repeated game also has a unique SPE [35]. Also observe that there are many sets of channel conditions that can lead to undesirable unique NEs. For instance, if $\alpha_i(g_2) = 0.92$, an undesirable unique SPE will ensue. In Section 3.4.7, we show that these situations are common using a realistic channel model for mobile environments.

A non-constructive proof of this theorem that gives insights on necessary conditions leading to undesirable SPEs under sDCF can be found in Appendix A.1.1.

The fundamental problem is that providing fixed long-term channel access probabilities while allowing variable channel time per transmission opportunity (as DCF does) leads rational nodes to use inefficient transmission strategies since they can increase their channel time shares by doing so.

3.4.5 Analysis of EDCF

In an attempt to provide QoS support for 802.11-based WLANs, an IEEE working group is drafting the 802.11e standard that specifies a distributed channel access protocol, EDCF, an enhancement to DCF. Unlike DCF, EDCF allows a node that wins the contention to transmit for a bounded interval of time t^{max} , irrespective of the frame size and data rate used. It appears that the main reason for limiting the duration of each TXOP is the predictability of the maximum frame transmission time, which is necessary to meet QoS guarantees. This limit also significantly affects the nature of competition.

EDCF, unlike DCF, allows bursts of frames to be transmitted. The maximum burst length depends on the data rate used. For instance, for $t^{max} = 7.35$ ms, at least five 1500-byte frames can be successively transmitted at 11 Mbps. However, at 5.5 Mbps, only about three 1500-byte frames can be transmitted. Like DCF, EDCF gives an equal long-term channel access probability (i.e., equal number of TXOPs) to competing nodes that have the same priority. However, the actual average number of

frames transmitted by a node in a transmission opportunity depends in part on the backoff scheme.

Distributed MAC protocols like DCF and EDCF employ a backoff scheme to resolve contention. Under DCF, after each frame transmission, a node picks a random number of $20\text{-}\mu\text{s}$ time slots between 0 and the contention window, CW , and remains idle during that backoff period. This allows another contender with a smaller backoff period to access the channel. Inevitably, frames sometimes collide and the number of collisions increases rapidly with the number of contenders. Like DCF, EDCF, uses an exponential backoff technique in which the contention window size is doubled for each failed frame transmission. If the previous frame transmission is successful, CW is set to a pre-determined minimum value, CW^{min} .

Under EDCF, a node can transmit multiple frames per transmission opportunity and any of those frames can be lost. The time at which a node backs off can affect the amount of channel time it gets. There are two major ways in which this can be done.

First, a node can stop transmitting subsequent frames as soon as it detects a failed transmission within the burst. We call this technique BFL (for Backoff upon First Loss). Since the wireless channel is lossy, the average of number of frames transmitted per transmission opportunity typically will be lower than the maximum allowed. Subsequently, the average channel time used per stagegame will be less than the maximum allowed, i.e., $t_i < t^{max}$.

Second, a node can transmit the maximum number of frames allowed regardless of failures, and back off only after the last frame transmission. We call this technique BEB (for Backoff at End of Burst). Under BEB, the average number of frames transmitted per transmission opportunity will be equivalent to the maximum allowed, i.e., $t_i = t^{max}$.

There are advantages and disadvantages to each technique. When there is only a single node transmitting, it is better to employ BEB since it increases the achieved throughput by reducing the total amount of backoff (idle) time. But, the backoff time cannot be reduced to zero since doing so will lead to a series of collisions when another nearby node arrives. Again, under CSMA, each node is required to listen to the channel and the probability of transmission of a node in each idle time slot is less than 1, leading to a positive amount of idle time between transmission events on average.

When multiple nodes are competing for channel access and losses are bursty, BFL is more desirable than BEB. In indoor mobile environments, channel conditions are time-correlated on short time scales because of multipath and mobility [83], and thus, whenever a frame transmission fails because of channel errors, it is likely that successive frame transmissions will also fail. Thus, under BFL, a node will avoid likely failed transmissions by backing off as soon as it experiences a frame loss. Meanwhile, a competing node with better channel conditions can transmit, improving the overall efficiency. It has been observed that the channel qualities of different transmission paths are often independent and thus losses on a single path are often bursty in mobile environments [71, 87]. As we explain shortly, EDCF with BFL leads rational nodes to use inefficient equilibrium strategies but EDCF with BEB does not. Again,

because we make the same simplifying assumptions for EDCF as we did for DCF (see Section 3.4.4), we call our EDCF version of sEDCF.

Lemma 3.4.4. *Under sEDCF, $t_i = \frac{b_i * s_i}{\gamma_{theo}(g_i)}$ and $t_j = \frac{b_j * s_j}{\gamma_{theo}(g_j)}$.*

Proof. The channel time allocated to node i under sEDCF during each stagegame is simply the amount of channel occupancy time needed to transmit the average number of frames (b_i) transmitted in each stagegame. □

Theorem 3.4.5. *Under sEDCF with BFL, there exists undesirable unique SPEs.*

Proof by construction. We assume the same set of channel conditions for two nodes i and j as the one we use in Table 3.3. Again, g_{11} and $g_{5.5}$ are the only two interesting data rates since the success rates of node j drastically differs under them. Assume that $t^{max} = 12.3$ ms. Thus, each node can transmit a maximum of 9 frames using g_{11} and a maximum of 5 frames using $g_{5.5}$, i.e., $n_i(g_{11}) = 9$ and $n_i(g_{5.5}) = 5$.

Node	$\alpha(g_{11})$	$\alpha(g_{5.5})$	$\gamma^{prac}(g_{11})$	$\gamma^{prac}(g_{5.5})$	$b(g_{11})$	$b(g_{5.5})$
i	0.63	0.99	4.404	3.824	2.29	4.90
j	0.99	0.99	6.921	3.824	8.648	4.90

Table 3.5: Frame success rates and corresponding practically achievable throughputs of node i and j at all possible transmission strategies under sEDCF with BFL.

Under sEDCF with BFL, b_i is the expected number of transmissions in each stagegame and depends on the transmission strategy used and the channel conditions experienced. $b_i(g_x)$ can also be computed from the overall frame success rate, α_i provided that frame losses are uncorrelated (as we have assumed). Let $p_i(k)$ be the probability of node i transmitting exactly k frames (i.e., the first $k - 1$ frames are successful and the last frame was a loss) in each stagegame where $0 < k < n_i$ and $n_i > 1$.

$$p_i(k) = \alpha_i^{(k-1)} * (1 - \alpha_i)$$

Note that $p_i(n_i) = \alpha_i^{(n_i-1)}$ since the n_i^{th} frame will be transmitted so long as the preceding $n_i - 1$ frames were successful.

Using $p_i(k)$ we can compute b_i as follows:

$$b_i = \sum_{k=1}^{n_i} (p_i(k) * k)$$

Table 3.5 lists b_i and b_j when each node is using either g_{11} or $g_{5.5}$.

Based on Lemma 3.4.4 and Equations 3.5, 3.4 and 3.3, we can compute all possible outcomes of each stagegame in Table 3.6. We make two observations. First, there exists a unique NE in which node i plays strategy $g_{5.5}$ and node j plays strategy g_{11} ; this unique NE is undesirable since $\gamma_i^{prac}(g_{11}) > \gamma_i^{prac}(g_{5.5})$. Second, except when both

g_j	g_{11}	$g_{5.5}$
g_i	$(f_i(G_I), f_j(G_I))$	$(f_i(G_I), f_j(G_I))$
g_{11}	(0.209, 0.791)	(0.205, 0.795)
$g_{5.5}$	(0.506, 0.494)	(0.500, 0.500)
	$(\gamma_i(g_i), \gamma_j(g_j))$	$(\gamma_i(g_i), \gamma_j(g_j))$
g_{11}	(0.920, 5.475)	(0.903, 3.04)
$g_{5.5}$	(1.935, 3.419)	(1.91, 1.91)

Table 3.6: Fractions of channel occupancy time and achieved throughputs of node i and j when each node is using either g_{11} or $g_{5.5}$. The unique NE strategies for i and j are $g_{5.5}$ and g_{11} respectively.

nodes are using $g_{5.5}$ the amount of channel occupancy time allocated to each node is unequal. Therefore, under sEDCF with BFL, there exists undesirable NEs and time-based fairness is not achieved at NEs under certain conditions.

A non-constructive proof of this theorem that gives insights on necessary conditions leading to undesirable SPEs under sEDCF can be found in Appendix A.1.2.

Theorem 3.4.6. *Let g_i^* and g_j^* be the strategies of nodes i and j at a NE. Under sEDCF with BEB, g_i^* and g_j^* are desirable strategies. I.e., any NE arrived under sEDCF with BEB is desirable. Since any NE is desirable, any SPE will also be desirable.*

Intuitively, the theorem states that if the system allocates the same amount of channel time regardless of the strategy used, each node at equilibrium will always use the strategy that yields the maximum practically achievable throughput.

Proof. We prove by contradiction. Suppose that there exists a strategy, $g'_i \neq g_i^*$, such that $\gamma^{prac}(g'_i) > \gamma^{prac}(g_i^*)$. Since $t_i^* = t^{max}$, $t'_i \leq t_i^*$. Thus, according to Equation 3.4, $\gamma_i(g'_i, g_j^*) > \gamma_i(g_i^*, g_j^*)$. But this contradicts the fact that g_i^* is the optimal equilibrium strategy for node i , given that node j uses g_j^* . A similar argument can be made for g_j^* being a desirable strategy.

Though the equilibrium is desirable, sEDCF with BEB can lead to higher overall frame loss rates than sEDCF with BFL. In other words, the aggregate throughputs achieved in a SPE can be improved if the MAC protocol provides flexibility. This is the topic of the next subsection.

□

3.4.6 Providing Flexibility at the MAC Layer Improves Performance

In this section, we demonstrate that the MAC layer can help rational nodes achieve more efficient equilibriums than those achieved under sEDCF with BEB, in the presence of time-correlated channel errors.

We now extend our error model. Let K be the number of stage games. For the rest of this section, we assume that $f^{chan} = 1$. Then, we have $2K$ burst transmission slots (BTSs), each of which lasts for t^{max} . Each node will be given K TXOPs to transmit in K BTSs. BTSs are numbered from 1 to $2K$. We represent the channel state of node i at the k^{th} BTS with a function $cs_i(k, g)$, where $1 \leq k \leq 2K$. $cs_i(k, g) = 1$ if all frame transmissions during the k^{th} BTS using transmission strategy g will be successful and $cs_i(k, g) = 0$ otherwise. Although in practice, a partial number of frame transmissions may fail or succeed within each BTS, for simplicity, we make an assumption that all fail or succeed with each BTS. The results in this subsection can be extended for cases when this assumption does not hold.

In the previous section, losses are random. Therefore, the best strategy for a node is to use the transmission rate that maximizes the practically achievable throughput (γ^{prac}) given the steady-state bit success rate. Here, we assume that it is common knowledge that losses occur in bursts. That is if $cs_i(k, g) = 0$, then it is highly likely that $cs_i(k + 1, g) = 0$. Therefore, each rational node will attempt to estimate the channel conditions and choose an appropriate transmission strategy for each BTS. Under the game model presented in Section 3.4.1, each node must transmit in the BTS granted to it. We now describe a new MAC protocol called *Flex-1*. For the rest of this subsection, we assume that Flex-1 uses BEB. Flex-1 provides a replacement BTS to a node that intentionally gives up its BTS. Specifically, we model Flex by extending our game model as follows. Therefore, in some cases, a node may decide to exchange its BTS for another node's BTS in the future. We redefine a transmission strategy g_i of node i as a triplet $(r_i, s_i, tx(k))$, where $tx(k)$ is a function that maps to 1 if node i 's transmission is highly likely to be successful in the k^{th} BTS and 0 otherwise.

As before, at the beginning each stage game, with probability $p = 0.5$ node i is designated to communicate first and with probability $1 - p$ node j is designated to communicate first. However, under Flex-1, the actual order of transmissions may change. Without loss of generality, assume that node i is selected to transmit at the beginning of a stage game at the k^{th} BTS. Flex-1 operates as follows during the stage game:

1. If node i transmits in the k^{th} BTS, Flex-1 will designate node j to transmit in the $k + 1^{th}$ BTS,
2. If node i decides not to transmit, Flex-1 will designate node j to transmit in the k^{th} BTS,
 - (a) If node j declines to transmit in the k^{th} slot, Flex-1 will again designate node i to transmit in that BTS and irrespective of node i 's decision, Flex-1 will designate $k + 1^{th}$ BTS to node j , and
 - (b) If node j transmits in the k^{th} BTS, then Flex-1 will designate node i to transmit in the $k + 1^{th}$ BTS.

For example, assume that node j transmitted in the $k - 2^{th}$ BTS and node i transmitted in the $k - 1^{th}$ slot. Now, node i is assigned to transmit in the k^{th} BTS.

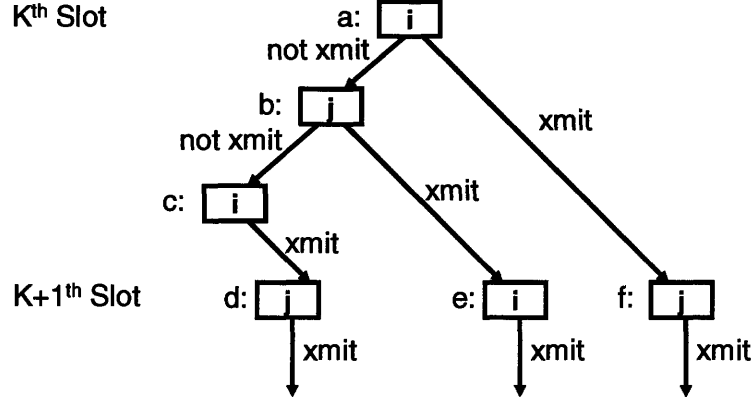


Figure 3-1: Decision tree of a stage game under Flex-1. Node i may choose not to transmit in which case node j may transmit. If node j does transmit, node i then transmits in node j 's turn.

Figure 3-1 describes a complete game decision tree involving a stage game starting at the k^{th} BTS when i and j are employing g_i^* and g_j^* respectively. We now prove the properties of Flex-1.

Definition 3.4.7. Poor channel conditions are *coherent* for T seconds if such conditions remain relatively unchanged for that duration. I.e, if transmissions of node i using $g_i = (r_i, s_i)$ fail in the k^{th} BTS, transmissions of i using the same g_i will be highly likely to fail in the $k + 1^{th}$ BTS. However, the success probability of i 's transmissions during the $k + 2^{th}$ BTS is independent of that during the k^{th} BTS.

Theorem 3.4.8. Let g_i^{single} and g_j^{single} be the maximally efficient strategies that nodes i and j will be using if they alone occupy the channel. Let g_i^* and g_j^* the strategies of nodes i and j at a NE under Flex-1 with BEB. g_i^* and g_j^* are no less (sometimes more) desirable than g_i^{single} and g_j^{single} . given that channel conditions are coherent for $2 * t^{max}$ seconds.

Proof. Let tx_i^{single} and tx_j^{single} be the functions that i and j use when they alone occupy the channel. Clearly, when a node is alone, it will always transmit at its assigned BTS. Therefore, $tx_i^{single}(k) = 1$ and $tx_j^{single}(k) = 1$ for any k .

We now prove by constructing g_i^* . Let $g_i^* = (r_i^{single}, s_i^{single}, tx_i^*)$. For every k^{th} BTS allocated to i by Flex-1, $tx_i^*(k)$ operates as follows. i may delay its transmission if i) it has transmitted in the $k - 1^{th}$ BTS, ii) transmissions failed during the $k - 1^{th}$ BTS, and iii) node j will transmit in this k^{th} BTS. Otherwise, node i transmits in the k^{th} BTS. g_j^* can be constructed in the same manner.

When all three conditions are not met, node i will always transmit in its designated slot using r_i^{single} and s_i^{single} . Therefore, $g_i^* = g_i^{single}$ in those BTSs.

If all three conditions are met, node i will delay its designated TXOP at the k^{th} BTS and transmit in the $k + 1^{th}$ BTS (allocated to it by Flex-1). For each instance in which all three conditions are met, node i will be able to transmit b_i frames in the

$k + 1^{th}$ BTS. The failure probability of transmissions during the $k + 1^{th}$ BTS equals the steady state loss probability and is less than that during the k^{th} BTS.

Also, node j had last transmitted in the $k - 2^{th}$ BTS, exactly 2 BTSs ago (since node i had transmitted in the $k - 1^{th}$ BTS). According to our assumed common knowledge on the loss process, irrespective of successes or failures of j 's transmissions during the $k - 2^{th}$ BTS, the failure probability of j 's transmissions in the k^{th} BTS is exactly the steady-state loss rate. In other words, by exchanging BTS with j , i has increased its success probability without increasing the failure probability of node j .

In Figure 3-1, the sequence a, b, e will denote the resulting actions of equilibrium strategies for the stage game, provided that i 's transmissions in the $k - 1^{th}$ BTS failed.

□

Flex-1 is a special case of *Flex-k*, a MAC protocol that allows a designated node to defer up to k transmission opportunities without losing them. We call k the *degree of scheduling flexibility*. If a rational node chooses not to transmit in more than k BTSs, it will not be offered a replacement BTS. Otherwise, the node can swap its BTS with another node. In practice, it is hard to keep track of the history of BTS usage of each node and designate BTSs according to each node's desire to transmit. However, a protocol like Flex-1 can be implemented in a distributed fashion by adaptively adjusting transmission probability of each node depending on i) how much channel occupancy time it used in the past and ii) its desired share of channel occupancy time. We describe such a practical and effective protocol in Chapter 5. The next section shows through simulation that under DCF, rational nodes can employ inefficient transmission strategies.

3.4.7 Evidence of Inefficient Equilibria Through Simulations

We conduct simulation runs in *ns* [76], relaxing some of the simplifying assumptions made in our analytical model.

Environments

We use a Rayleigh fast-fading model [80, 83] to capture the short time-scale fading phenomenon that arises because of objects moving along the transmission path between a transmitter and a receiver, which may also be moving. The received power thresholds for various data rates are based on the Orinoco 802.11b Gold Card data sheet.

Unlike in the previous section, we use TCP instead of UDP, to demonstrate that our results apply to TCP. In our analytical model, we also assumed that the channel conditions in each subgame are constant, leading to each node transmitting at the most appropriate transmission rate for the entire duration, given that all other nodes choose their best transmission rates. In practice, channel conditions vary and wireless card vendors employ proprietary auto-rate adaptation schemes that adjust the data transmission rate (on a frame-by-frame basis) based on estimated channel conditions.

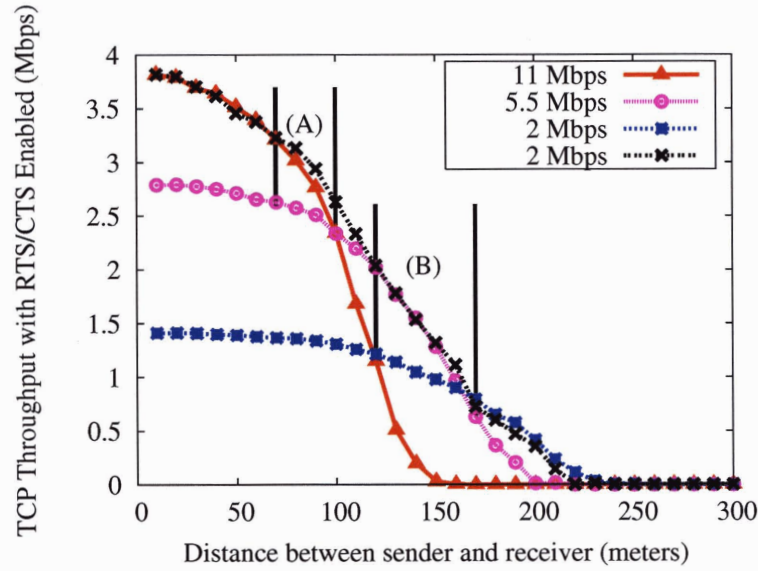


Figure 3-2: TCP throughput achieved when using various fixed data rates and RBAR, an auto-rate protocol. Regions (A) and (B) are where rational nodes under DCF may use inefficient strategies when competing against nodes with lower loss rates (smaller transmission distances).

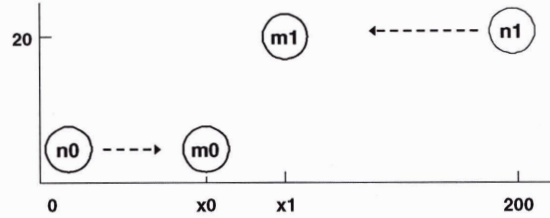


Figure 3-3: n_0 and n_1 transmit to m_0 and m_1 respectively.

Our simulation takes into account auto-rate protocols. Our analytical results agree with the simulation results despite the differences.

For concreteness in our examples, we use the Receiver-based Auto-rate protocol (RBAR) [42]. An RBAR receiver informs a sender of channel conditions before the sender transmits a data frame. In particular, the sender sends an RTS (request to transmit) frame and the receiver reports the received signal strength of the RTS frame in a replying CTS (clear to transmit) frame. The RTS/CTS scheme is typically used to reduce collisions as a result of frame transmissions by hidden nodes. Compared to data frames, the RTS and CTS frames are very small and are transmitted at 2 Mbps making them robust against channel errors. Based on the signal strength information, the sender then chooses the highest transmission rate at which successful frame transmission is highly likely, under the assumption that the channel conditions will remain unchanged for the transmission period. Figure 3-2 shows that in most cases RBAR performs well as it adapts the transmission rate based on observed

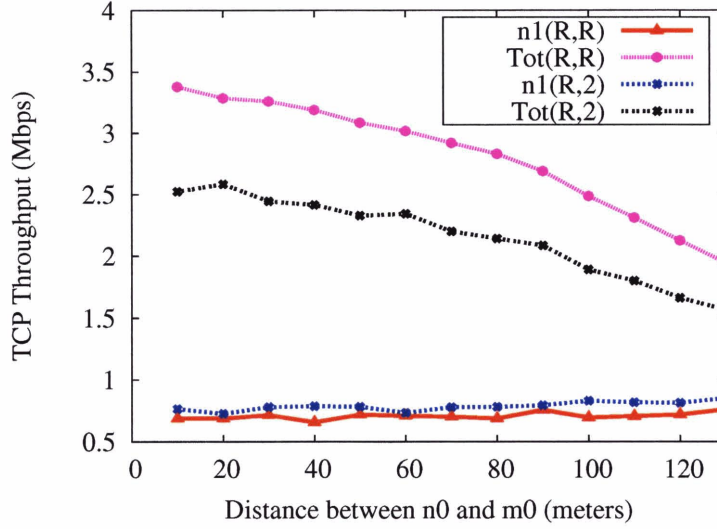


Figure 3-4: TCP throughput achieved by $n1$ and the aggregate achieved throughputs under two pairs of strategies as a function of the distance between $n0$ and $m0$. $(R, 2)$ denotes that $n0$ uses RBAR and $n1$ transmits at a fixed data rate of 2 Mbps. $Tot(R, 2)$ plots the aggregate throughputs. However, the most efficient strategy for $n1$ is to transmit at 5.5 Mbps, which is what RBAR running at $n1$ would do. Thus, (R, R) denotes the most efficient strategy pair which may not be used at equilibriums.

channel conditions.

However, a rational node may not choose its transmission strategy solely based on its channel conditions. A rational node should periodically evaluate its achieved throughput, channel conditions, observed channel time usage and average frame loss rate to determine the best strategy for transmitting data frames. As evident in our analyses in the previous sections and this section, the transmission strategy that maximizes the achieved throughput of an individual node is not necessarily the most efficient one.

Results

We ran experiments using the setup shown in Figure 3-3. There are two TCP flows, one from $n0$ to $m0$ and the other from $n1$ to $m1$. $m0$ and $m1$ also send TCP acknowledgment packets to $n0$ and $n1$ respectively. The positions of $n0$, $m1$, and $n1$ were fixed whereas that of $m0$ was varied. $m1$ was 130 m away from $n1$ (i.e., $x1 = 70$ m), and the distance between $n0$ and $m0$ was varied from 10 to 130 m. All nodes are within radio transmission range of each other.

We also ran a set of experiments using UDP flows. The results were similar in nature and since TCP is more widely used, we only include the results for TCP experiments.

When both nodes used RBAR, $n1$ achieved lower throughput than $n0$ when its distance from $m1$ was farther than that between $n0$ and $m0$. The most efficient data

rate for $n1$ would be 5.5 Mbps if $n1$ had the channel all to itself (see Figure 3-2). This was what RBAR did most of the time. However, in the presence of a competing flow, $n1$ could achieve higher throughput by transmitting at 2 Mbps. This behavior is evident in Figure 3-4 which shows the achieved throughputs of $n1$ and the aggregate throughputs as a function of the distance between $n0$ and $m0$. For example, when $m0$ is 10 m away from $n0$, $n1$ can achieve an 11% increase in throughput by always transmitting at a lower data rate instead of using RBAR. However, as a result of $n1$ using this inefficient strategy, the achieved throughput of $n0$ (not shown in the figure) and the aggregate throughput would decrease by 53% and 34% respectively.

In Figure 3-4, $n1$ only gains an 11% increase in throughput by transmitting at a less efficient rate of 2 Mbps instead of transmitting at 5.5 Mbps. However, the figure only shows an example scenario illuminating the impact of arriving at inefficient equilibriums under DCF. There are certainly cases where $n1$ could gain much higher throughputs by transmitting at inefficient data rates at the expense of reducing aggregate throughputs.

We ran numerous experiments to determine the regions in which rational nodes could benefit by transmitting at an inefficient data rate. In Figure 3-2, a node in region A or B can achieve higher throughput by choosing a data rate lower than the most efficient data rate, whenever it competes against node that experiences a lower loss rate. The wide ranges of regions A and B highlight the importance of incorporating mechanisms to reduce inefficiencies as a result of competition among rational nodes in non-cooperative environments.

The simulation results (not shown here) for EDCF with FLB are similar to those described here although the regions where rational nodes may use inefficient strategies under EDCF are smaller than those under DCF.

3.4.8 Discussion

In this section, we discuss the relevance of our analysis in today's world of 802.11 WLANs. For each 802.11 wireless interface card, there are two functional components: *standard-compliant and customizable*. The *standard-compliant* component includes implementations (usually in firmware) of MAC and physical layers that are compliant with IEEE 802.11 specification. Thus, parameters such as CW should be set according to the specification. In practice, each 802.11 product undergoes a certification process administered by the Wi-Fi Alliance, a nonprofit international association formed in 1999 to certify interoperability of WLAN products based on IEEE 802.11 specification [103]. Presumably, the certification process will verify whether a product is compliant with the specification. Assuming that 802.11 wireless interface manufacturers want a wide-acceptance of their products by being standard-compliant and certified products, there is little incentive for them to improve performance of their products in a way that violates the specification. For instance, in theory, a node may opt to transmit frames without backing off, i.e., set $CW = 0$ or set it to a small value. But a rational manufacturer may not do that with for fear its products not being certified.

On the other hand, each manufacturer or even user can customize MAC layer

related parameters that are left unspecified by the standard. Data rate and frame size as part of the *customizable component* since 802.11 specification does not limit how such parameters are used. In practice, each card manufacturer often has its own proprietary auto-rate protocol to choose an appropriate data rate for each frame transmission, as we mentioned before. Furthermore, users can also adjust those parameters by modifying publicly available software drivers that act as the interface between the private firmware implementation of the MAC protocol and the networking stack of the operating system. As we have demonstrated throughout this section, enhancements to the 802.11 MAC protocol is necessary to prevent rational nodes from arriving at inefficient equilibria by modifying customizable parameters such as data rate. Chapter 5 describes a new distributed MAC protocol that provides long-term time share guarantees, thereby leading rational nodes to efficient equilibria.

Chapter 4

Comparison of Fairness Notions

In this section, we describe various measures for comparing pairs of fairness notions and use them to compare various fairness notions, including TF and BF. We also report on simulations that validate our analysis.

Let A and B be two fairness notions. Assume that both fairness notions share a common definition of entities. So, A and B only differ on their definitions of fairness units. Let:

u_i : the utility function of entity i .

u_i maps to a non-negative number. For example, the utility of i could be its achieved throughput or average task wait time. We will use U^F to denote the vector of achieved utilities under F , where $u_i^F \in U^F$ is the achieved utility of entity i under F .

4.1 Measuring Differences in Performance

We define $AggrDiff(A, B)$ as the aggregate utility gain or loss achieved under fairness notion A over that achieved under fairness notion B , as a fraction of the aggregate utility achieved under B :

$$AggrDiff(A, B) = \frac{\sum_{i \in I} u_i^A - u_i^B}{|\sum_{i \in I} u_i^B|} \quad (4.1)$$

$AggrDiff(A, B) > 0$ if there is a gain in aggregate utility under A over under B . Similarly, $AggrDiff(A, B) < 0$ if there is a loss in aggregate utility under A over under B .

4.2 Comparing Trade-offs between Performance and Relative Fairness

Fairness is a subjective notion. At the same time, we cannot ignore the performance impact of each fairness notion. It is often useful to understand how one fairness notion

differs from another in terms of the degree of “relative unfairness” and the aggregate achieved utility. In that aspect, we describe a quantitative measure, the *PF ratio*, that attempts to reflect the advantages and disadvantages.

Our goal is not to get into a philosophical argument on which fairness notion is “fairer”. We also set aside the questions we raised in an earlier chapter: what are the fairness units and what are the entities. Instead, we solely focus on the achieved utilities of entities under a pair of fairness notions and attempt to develop a measure that will capture the differences in the aggregate utility and the utility of each entity under one fairness notion from the other.

Traditional fairness measures such as Jain’s fairness index [46] only quantify how far a particular allocation is from the desired one and not address the relations between relative fairness and performance.

We begin with a simple example to illustrate the difficulties in developing such a measure that captures the trade-offs between relative fairness and performance. Our discussion in this section solely focuses on the achieved utilities of entities (not fairness units) under each fairness notion considered.

Let $U^A = \{0.5, 2\}$ and $U^B = \{0.4, 1.2\}$ be the utilities achieved by entity i and entity j under a fairness notion, A , and a fairness notion, B , respectively. From the performance standpoint, notion A leads to a higher aggregate utility than notion B . $AggrDiff(A, B) = 0.56$. Specifically, compared to B , A leads to increased utility for *all* entities. In this particular case, most, if not all, system designers and users would prefer A over B . We believe that if the performance of every entity is *no less* under A than under B , it is not meaningful to even talk about relative fairness since every entity under A achieves at least the utility that it would achieve under B .

Similarly, we are also not interested in comparing fairness notions when the aggregate utility under both notions is the same (but the utility of each entity under one notion may differ from that under another notion). In such a case, the advantages and disadvantages of that one fairness notion has over another cannot be discussed in the context of aggregate utility (which is the same).

We are, however, interested in understanding relative fairness and its relationship to aggregate utility gain when i) only some entities (not all) achieve higher utilities under one notion than the other and ii) the aggregate utility is higher under one notion. Let $U^C = \{0.2, 2\}$ be the utilities achieved by entity i and entity j under C . Now, compared to B , entity i achieves less utility under C but entity j achieves a higher utility. Observe that the total achieved utility of 2.2 is higher under C . Thus, we can say that C yields higher performance than B but some entities under C achieve less utility than they would achieve under B . In our opinion, *relative unfairness* between two fairness notions can be captured with relative loss of utility.

We present an intuitive way to quantify the relationship between relative fairness and relative performance of two different fairness notions. Let Tar (for target) and Ref (for reference) be two fairness notions and U^{Tar} and U^{Ref} be the utility vectors under Tar and Ref respectively. We require that the aggregate utility under Tar is different from that under Ref .

We define the relative performance to fairness ratio (or PF) of notion Tar over

reference notion Ref as:

$$\begin{aligned}
 PF(Tar, Ref) &= \frac{\text{Difference in the aggregate utility under } Tar \text{ over that under } Ref}{\text{Aggregate utility loss under notion } Tar \text{ over notion } Ref} \\
 &= \frac{\sum_{i \in I} u_i^{Tar} - u_i^{Ref}}{\sum_{i \in I} \max(0, u_i^{Ref} - u_i^{Tar})} \quad (4.2)
 \end{aligned}$$

The numerator quantifies the total performance gain of using notion Tar over notion Ref . The denominator quantifies the relative unfairness, of using Tar over Ref .

In the previous example, the PF of notion C over notion B is: $PF(C, B) = \frac{0.2-0.4+2-1.4}{0.4-0.2} = 2$. One way to interpret this result is that for every unit of utility lost by some entity under notion C , there are two units of net aggregate utility gained by other entities under notion C .

4.3 Throughput Comparison

In this section, we compare achieved throughput under different notions of fairness based on the analysis presented in Section 3.2 and through simulation. Although our throughput analysis in Section 4.3 is for raw MAC-layer achieved throughput, we can easily adapt it to understand the impact of MAC-layer fairness notions on both achieved UDP and TCP throughput. For the rest of this chapter, we consider that competing entities are links (not nodes).

We assume that each UDP packet will fit in a single MAC-layer frame. The achieved UDP throughput of each node under a fairness notion A can be computed as:

$$\gamma_i^{A,udp} = \frac{s_i - b^{udpovh}}{s_i} * \gamma_i^A$$

where $s_i - b^{udpovh}$ represents the UDP payload bits per frame, b^{udpovh} is the combined size of UDP and IP headers in bits (224 bits), and γ_i^A is given in Section 3.2.4.

We can also compute achieved TCP throughput in a similar manner. We assume that each TCP data packet will fit in a single MAC-layer frame. Under per-link, bit-based fairness, each client exchanging TCP data packets will be able to transmit an equal number of TCP data bits. Similarly, under per-link, time-based fairness, each client exchanging TCP data packets will be able to transmit for an equal amount of channel occupancy time. Unlike UDP flows, TCP data flows have corresponding TCP *ack* flows in opposite directions. Therefore, when computing achieved TCP throughput, we must consider the overhead incurred by TCP *ack* packets. The achieved TCP throughput of entity i under fairness notion A can be computed as follows:

$$\gamma_i^{A,tcp} = \frac{s_i - b^{tcpovh}}{s_i} * (1 - f^{tcpack}) * \gamma_i^A$$

where b^{tcpovh} is the combined size of TCP and IP headers in bits (320 bits), γ_i^A is given in Section 3.2.4, and f^{tcpack} is the fraction of channel occupancy time used for transmitting TCP *ack* packets relative to the total amount of channel occupancy

time.

Data Rate	γ^{theo}	ϕ^{BF}	ϕ^{TF}
11	8.7625	0.06034	0.25
5.5	4.8377	0.1093	0.25
2	1.8865	0.2803	0.25
1	0.9614	0.5500	0.25

Table 4.1: The theoretically achievable MAC-layer throughput (in Mbps) and the fair share of channel occupancy time of an entity (ϕ) running at each of the four possible 802.11b data rates (in Mbps).

For the rest of this chapter, we will compare the throughput and delay achieved under TF and BF. BF allocates a lesser share of channel occupancy time to a faster node than a relatively slower node (see Equation 3.12). TF allocates equal shares of channel occupancy time among nodes 3.13. Naturally, this raises a question of potential fairness notions that allocate a bigger share of channel occupancy time to a faster node than a relatively slower node. We will discuss such fairness notions in Section 4.7. Until then, we focus on comparing BF and TF.

Table 4.1 lists the theoretically achievable MAC-layer throughput and fair share of channel occupancy time of an entity at each of the four possible 802.11b data rates under TF and BF. We examine a simple scenario, in which two nodes, 11 and 1, which transmit UDP data packets at 11 and 1 Mbps respectively to different receivers. Unless otherwise noted, for the rest of this chapter, we assume that no frames are lost because of channel errors. We run *ns-2* simulations to obtain the achieved throughput of each node under both BF and TF, in the absence of channel errors. Since both nodes use the same frame size, BF and FF, achieved by DCF, are equivalent. To achieve TF, we manually configure the contention window of each node so that the amount of channel occupancy time allocated to each node is equal. We also disabled DCF's exponential backoff algorithm to ensure fairness. For each experiment, we obtained the overall success rates of competing nodes, α_{11} and α_1 . We also obtained f^{chan} , the fraction of channel occupancy time required for all frame transmissions relative to the duration of the competing period.

Table 4.2 lists the simulation results under TF and BF using the modified DCF. Under BF, $\alpha_{11} = \alpha_1$ since the fraction of frame transmissions that are involved in collisions is the same for both nodes. Similarly, the achieved UDP throughput of each node is equal. The achieved UDP throughput obtained through simulation and the analytical results using Equation 3.11 are identical. These equations contain variables that are not derived: α_{11} , α_1 and f^{chan} . For those variables, we substitute the values obtained from the simulation in Equation 3.11. The values of these variables are MAC-protocol specific and in Chapter 5, we derive these values analytically for our proposed MAC protocol, TES.

Under TF, α_{11} is significantly higher than α_1 . This is because under TF, node 1 transmits about 9 times fewer frames than node 11. Therefore, the frequency of

Notion	α_{11}	α_1	f^{chan}	$\frac{s_i - b^{udpovh}}{s_i}$	γ_{11}^{UDP}		γ_1^{UDP}	
	(sim)	(sim)	(sim)	(sim)	(sim theo)	(sim theo)	(sim theo)	(sim theo)
BF	0.9376	0.9376	0.9774	0.9813	0.779 0.779	0.779 0.779	0.779 0.779	0.779 0.779
TF	0.9664	0.6937	0.9761	0.9813	4.056 4.055	4.056 4.055	0.319 0.319	0.319 0.319

Table 4.2: Simulation results and analytically derived UDP throughputs of an experiment in which node 11 sends UDP data packets at 11 and node 1 at 1 Mbps.

node 1 frames colliding with another node sending more frames (node 11) than it is higher than that node. So, node 1 suffers a significantly higher loss rate attributable collisions than node 11. Again, the achieved UDP throughput of each node under simulation and analysis is nearly identical.

We now compare TF and BF using the measures developed in the previous section. Using Equation 4.1, we can compute the fraction of the difference in aggregate utility under fairness notion A to that achieved under B as:

$$AggrDiff(A, B) = \frac{\gamma_{11}^A + \gamma_1^A - (\gamma_{11}^B + \gamma_1^B)}{\gamma_{11}^B + \gamma_1^B}$$

Similarly, using Equation 4.2, we can compute the PF ratio, $PF(A, B)$, as follows:

$$PF(A, B) = \frac{\gamma_{11}^A + \gamma_1^A - (\gamma_{11}^B + \gamma_1^B)}{\gamma_1^B - \gamma_1^A}$$

Measure	(TF, BF)
AggrDiff	1.80
PF	6.12

Table 4.3: $AggrDiff$ and PF ratios for UDP experiments.

Table 4.3 describes $AggrDiff$ and PF of (TF, BF). As shown in the table, TF improves aggregate UDP throughput over BF by 180%.

The ratio, $PF(TF, BF)$, is high suggesting that a significant improvement in aggregate throughput can be realized under TF over BF. For example, for each unit of UDP throughput lost by node 1 under TF (compared to under BF), node 11 gains that unit plus an additional 6.12 units of throughput.

We also examine a scenario, in which two nodes send TCP data packets to different receivers. Node 11 transmits TCP data packets at 11 Mbps and its receiver also transmits TCP *ack* packets at 11 Mbps. Similarly, both node 1 and its receiver transmit TCP data and *ack* packets at 1 Mbps. Table 4.4 shows the simulation results along with analytically derived TCP throughputs. $(1 - f^{tcpack}) * f^{chan}$ is larger under BF than under TF. This is because the fraction of channel occupancy time needed to transfer TCP *ack* packets is higher for the faster TCP flow than the slower

TCP flow; the transmission time of a TCP *ack* packet, irrespective of the data rate used, is dominated by a fixed amount of time required to transmit the physical layer preamble and header. Table 4.5 shows the *AggrDiff* and *PF* values for the TF and BF pair. The results are similar to those observed in UDP experiments. As a result of a higher fraction of TCP *ack* overhead for the faster TCP flow, *AggrDiff* of a particular pair of fairness notion is smaller for the TCP experiment than the UDP experiment.

Notion	α_{11}	α_1	$(1 - f_{tcpack}) * f_{chan}$	$\frac{s_i - b^{tcpowh}}{s_i}$	γ_{11}^{TCP}		γ_1^{TCP}	
	(sim)	(sim)	(sim)	(sim)	(sim theo)		(sim theo)	
BF	0.8243	0.8267	0.9100	0.9733	0.630 0.633		0.635 0.634	
TF	0.8224	0.7038	0.8392	0.9733	2.938 2.943		0.277 0.276	

Table 4.4: Simulation results and analytically derived TCP throughputs of an experiment in which node 11 exchanges TCP data with a receiver at 11 Mbps and node 1 exchanges TCP data with another receiver at 1 Mbps.

Measure	(TF, BF)
<i>AggrDiff</i>	1.54
<i>PF</i>	5.45

Table 4.5: *AggrDiff* and *PF* ratios between TF and BF for TCP experiments.

4.4 Delay Comparison

In practice, many WLANs experience congested periods interspersed with non-congested periods in which traffic loads do not exceed the channel capacity, as evident in our trace-driven analysis in Section 4.5. Session wait time of WLANs is a meaningful measure during congested periods when the wireless channel is the bottleneck for many flows.

In this section, we compare various pairs of fairness notions for two session classes, class 11, whose sessions are exchanging data at 11 Mbps and class 1, whose sessions are exchanging data at 1 Mbps. Both session classes have the same arrival process, whose inter-arrival time distribution is exponential with parameter λ . In the previous section, we analyze the impact of fairness notions on throughput in the context of all competing entities continuously sending data for the entire duration of the simulation period, i.e., the start and finish times of each entity are the same. However, in this section, each session only transfers a fixed amount of data. We only consider TCP sessions since most file transfer and web applications, whose utilities depend on session delay, use TCP.

Session Class	$\gamma^{prac,tcp}$ (Mbps)	μ (sessions per second)
11	5.696	2.422
5.5	3.144	1.337
2	1.226	0.5213
1	0.625	0.2657

Table 4.6: The practically achievable TCP throughput and the mean service rate, based on average session size of 294 KB, at each of the four possible 802.11b data rates. We assume that $\gamma^{prac,tcp}$ only depends on data rate, i.e., it remains the same under any fairness notion and any diverse mixes of data rate. We compute $\gamma^{prac,tcp}(d)$ by multiplying $\gamma^{theo}(d)$ with 0.65, i.e., the combined collision, idle time and TCP ack is assumed to be 35% of the run time.

We will make two simplifying assumptions in comparing the impact of various fairness notions on session delay when TCP is used: i) average loss rate (because of collisions) of each session remains the same, and ii) the term, $(1 - f^{tcpack}) * f^{chan}$, remains the same under various fairness notions. Our simulation results in the previous section showed that both assumptions are not true in some cases under a DCF-like MAC protocol, i.e., when TCP is used, the MAC-layer overhead, the combination of collision and idle time overhead, varies with fairness notion. However, the impact of differing collision rates and f^{tcpack} on achieved throughput and delay is less noticeable when compared to the impact of differing channel occupancy time allocation among nodes competing at different data rates. Therefore, for clarity and simplicity, we only examine the sole impact of various channel occupancy time allocations on session delay, with the assumption that the MAC-layer overhead remains the same under various fairness notions. Based on this assumption, the achieved TCP throughput of entity i under fairness notion A is:

$$\gamma_i^A(G_I) = \gamma_i^{prac,tcp}(g_i) * f_i^A(G_I)$$

where $\gamma_i^{prac,tcp}$ is the practically achievable throughput that remains the same under different fairness notions. The middle column of Table 4.6 lists $\gamma^{prac,tcp}$ for sessions running at four possible 802.11b data rates.

The service time distribution of each node depends on its session size distribution as well as the maximum achievable throughput. We assume that each class c session has the exponential session size distribution with mean session size of l_c bits. Thus, the service time distribution of class c sessions will also be exponential with $\mu_c = \frac{E[l_c]}{\gamma_c^{prac}}$. Based on our trace analysis of wireless network traces collected at Dartmouth College [60], we find that an average session size is 294 KB. Table 4.6 shows the mean departure rate, μ , in terms of sessions per second of four types of sessions, each running at one of four possible 802.11b data rates.

Let d_{aggr}^{max} be the maximum of the aggregate delay of competing nodes achieved

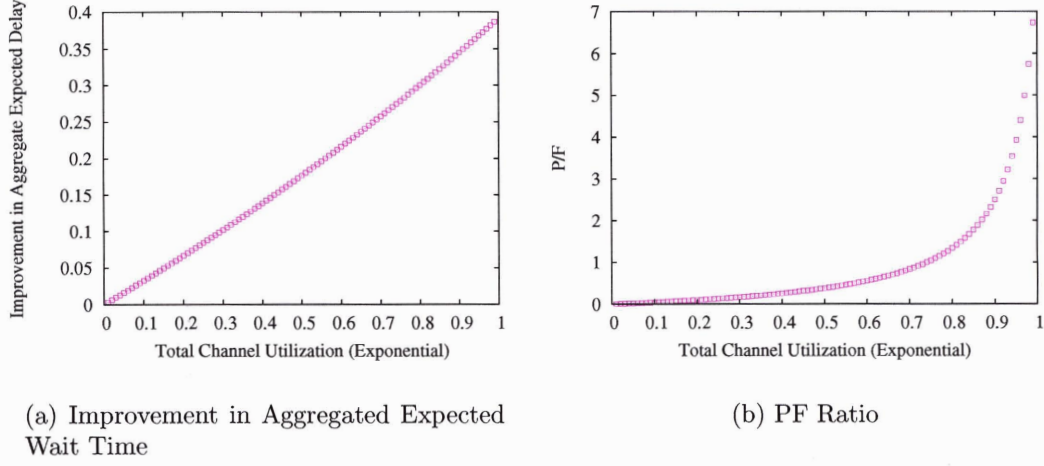


Figure 4-1: The improvement in aggregate expected wait time of all sessions achieved under TF over BF ($AggrDiff(TF, BF)$) and the PF ratio of TF and BF.

under fairness notions that are being considered. I.e.,

$$d_{aggr}^{max} = \max_F \left\{ \sum_{i \in I} E[d_i^F] \right\}$$

Let the utility of entity i under A be:

$$u_i^A = d_{aggr}^{max} - E[d_i^A]$$

That is the higher the delay, the lesser the utility. This notion will also lead to an intuitive result when used with $AggrDiff$, as we explain shortly.

Using Equations 4.1, 4.2 and 3.23, we now state the relative gain in aggregate utility, relative performance to fairness ratio and worst-case relative ratio of wait time of (A, B) , assuming that under A , node 11 is allocated more channel time than it would under B .

$$\begin{aligned} d_{aggr}^{max} &= E[d_1^B] + E[d_1^B] \\ AggrDiff(A, B) &= \frac{(-E[d_{11}^A] - E[d_1^A]) - (-E[d_{11}^B] - E[d_1^B])}{d_{aggr}^{max} - E[d_{11}^B] + d_{aggr}^{max} - E[d_1^B]} \\ PF(A, B) &= \frac{-E[d_{11}^A] - E[d_1^A] + E[d_{11}^B] + E[d_1^B]}{-E[d_1^B] + E[d_1^A]} \\ WorstIndivRatio(A, B) &= \frac{\phi_1^B}{\phi_1^A} \end{aligned}$$

Figure 4-1(a) plots $AggrDiff$ of the expected delay of TF over BF, as a function of channel utilization (ρ). TF can reduce expected wait time of sessions by as much as 39% over BF.

Figure 4-1(b) plots three PF ratios. $PF(TF, BF)$ increases with ρ at an increasing

rate. In other words, when the load is high, the expected wait time of slower sessions under TF is not much worse than that under BF while the net improvement of delay that TF has over BF is quite significant.

We also examine the relative worst-case wait time ratio, *WorstIndivRatio* (see Equation 3.23).

$$\begin{aligned} \text{WorstIndivRatio}(TF, BF) &= \frac{0.90}{0.5} = 1.8 \\ \text{WorstIndivRatio}(BF, TF) &= \frac{0.50}{0.1} = 5 \end{aligned}$$

WorstIndivRatio(TF, BF) is much lower than *WorstIndivRatio*(BF, TF). Based on the equation, it should be clear that under TF, *WorstIndivRatio* ≤ 2 , compared to any fairness notion. In general, under TF, *WorstIndivRatio* $\leq n$, where n is the number of competing sessions. However, the lesser the fraction of channel occupancy time allocated to the slowest node, the larger *WorstIndivRatio*. This is a strong reason why slower sessions should not be given much lesser shares of channel occupancy time than faster sessions, since doing so can lead to a large (unbounded) degree of unfairness. We will elaborate on this issue in Section 4.6. Allocating slower sessions much greater shares of channel occupancy time than slower sessions also leads to large *WorstIndivRatio*. The worst case relative wait time ratio of BF to TF (5) is much worse than that of TF to BF (1.8).

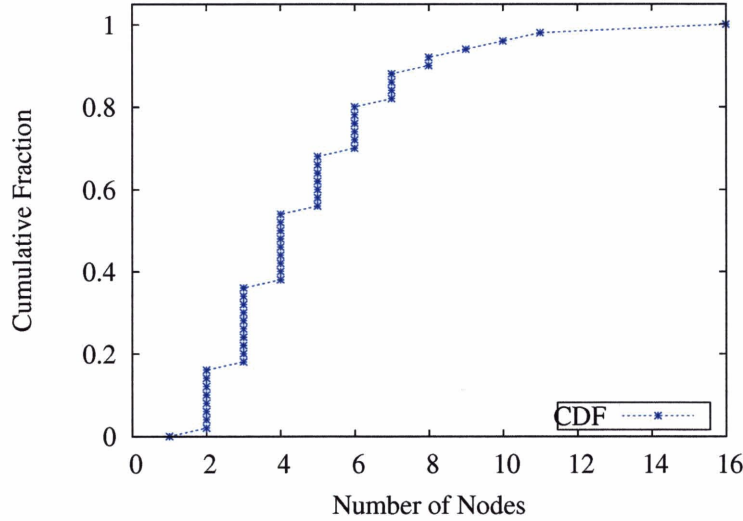


Figure 4-2: CDF of number of nodes during congested periods.

4.5 Trace-driven Analysis of Wait Time

In this section, we evaluate various channel time allocation policies on actual traffic loads. The results in the previous section predict that the higher the channel time allocated to faster nodes, the smaller the average aggregate wait time. In this section, we examine and answer two questions: i) how significant is the performance gain and ii) what is the penalty paid by slower nodes.

We analyzed wireless tcpdump trace of Whittemore, a residential facility in the Dartmouth business school where students were required to own laptops. This data was collected by Kotz *et al.* over the spring semester of 2002 [60] and was made publicly available. More than 90% of the bytes exchanged were TCP traffic. Unfortunately, the trace data does not contain the data transmission rate used for each frame transmission. Therefore we use the trace data to get actual distributions of session sizes and number of active nodes during periods of congestion, and then run a series of simulations using hypothetical distributions of transmission rates.

We define *congested intervals* as intervals in which the total data throughput at the AP exceeds 4 Mbps. This is 80% of the commonly observed TCP saturation throughput when nodes transmit at the maximum data rate and experience a loss rate of 1% to 2%. We identified congested periods in 300-ms windows. During each congested period, we identify the release time and size of each session. We then evaluate the wait time achieved by each session during each congested period for different channel time allocation policies. Each session is associated with a client exchanging data with another node. We assume that the same data rate is used in both directions.

We conducted 50 runs each with a different random seed to obtain a meaningful comparisons of various allocation policies. The results in this section are based on the data of all 50 runs, consisting of 175,900 sessions in 37,650 busy periods.

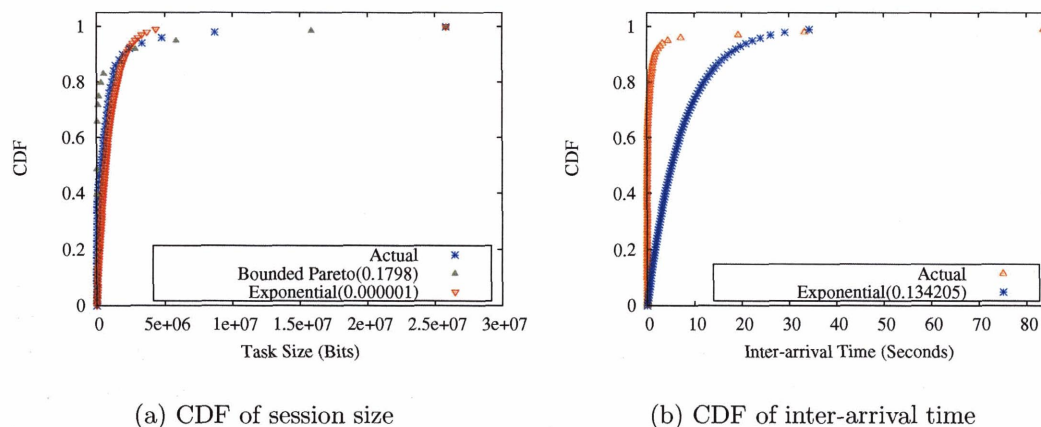


Figure 4-3: CDFs of session size and inter-arrival time of actual trace and other traffic distributions using the same mean values obtained from the trace.

4.5.1 Workload Characteristics

We present our results based on the trace collected on April 8, 2002 from 9 a.m. to 5 p.m. Figure 4-3(a) shows the cumulative fractions (CDFs) of session size from actual workload as well as the exponential and bounded pareto distributions, derived using the mean parameters obtained from the trace. As shown in the figure, 85% of the

sessions are less than 170 KB (or 1.4 Mb). However, the largest 5% of the sessions are greater than 560 KB. The actual size distribution seems to have the shape of an exponential size distribution. Similar statements can be said for inter-arrival times of sessions (see Figure 4-3(b)).

Figure 4-2 shows the CDF of number of active nodes during congested periods. The median number of nodes is 4. The size of each session in each congested period widely varies.

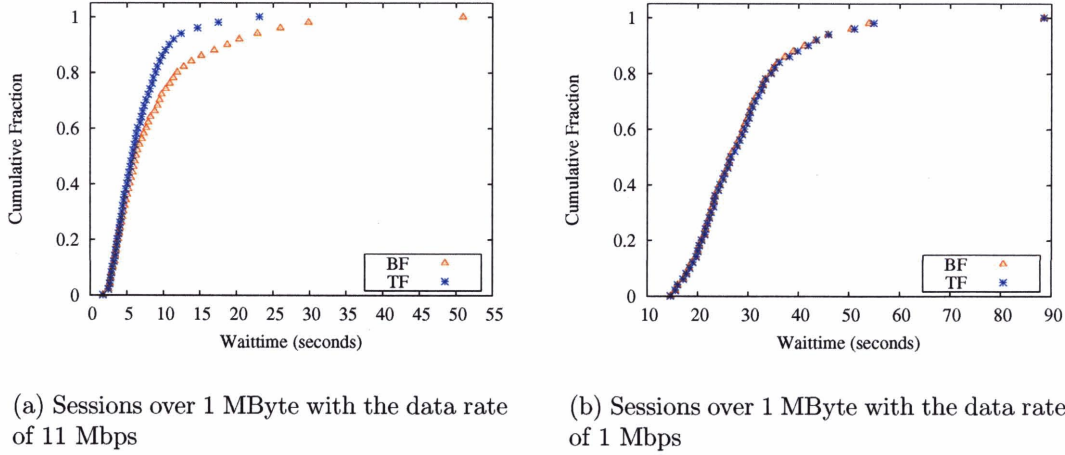


Figure 4-4: Distributions of wait time of 11 Mbps and 1 Mbps sessions. The data rate used by each session in this experiment is uniformly distributed among four 802.11b data rates.

4.5.2 Impact of Allocation Strategy on Wait Time

Since the trace data does not contain the data transmission rate used for each frame transmission, we consider three different possible distributions of transmission rates: i) a uniform distribution in which each frame is transmitted at a data rate among all possible rates with equal probabilities, ii) a distribution in which frames are mostly transmitted at the fastest speed, and iii) a distribution in which frames are mostly transmitted at the slowest speed.

We consider only 802.11b networks in which four different data rates are possible: 11, 5.5, 2 and 1 Mbps. The maximum achievable TCP throughputs using these data rates are 5, 3.3, 1.5 and 0.8 Mbps respectively. We examined sessions that are at least 100 KB in size (12,749 sessions) and sessions that are at least 1 MByte in size (897 sessions). The results in both cases are similar and we only report the results for sessions that are at least 1 MByte in size.

Figure 4-4 shows various subsets of the wait time distributions under the uniform distribution of 802.11b transmission speeds. Figure 4-4(a) shows the distribution of wait time for sessions that are at least as large as 1 MByte and use the fastest data rate of 11 Mbps. As expected, the observed wait time decreases with the increased

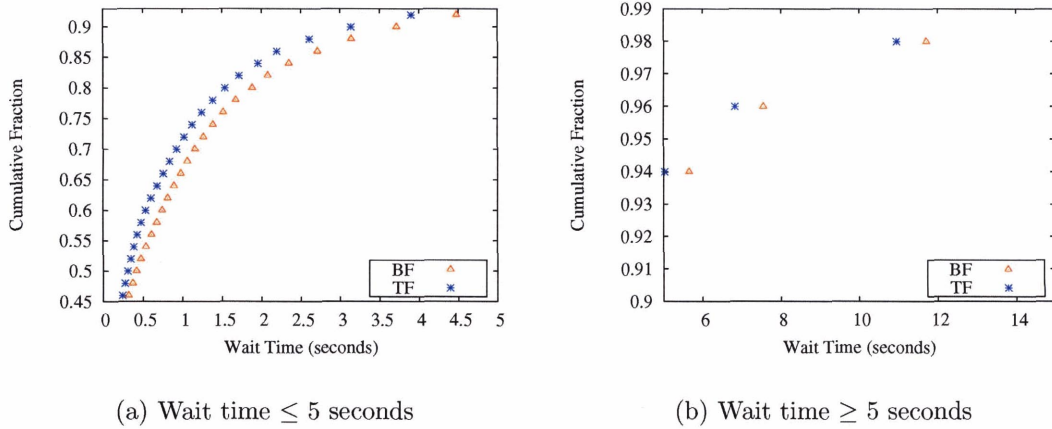


Figure 4-5: Distributions of wait time of all sessions. The data rate used by each session in this experiment is uniformly distributed among four 802.11b data rates.

allocation of channel time to the fastest nodes. The median wait time of nodes with the fastest speed under TF is 8.3% smaller than that that achieved under BF. Observe that the improvement is more significant at very large session sizes. For instance, the 90-percentile wait time of the sessions using the fastest data rate is 42.4% lower under TF than BF.

Figure 4-4(b) shows the distribution of session wait time for sessions that are at least as large as 1 MByte and use the slowest data rate of 1 Mbps. Here the median wait time of the slowest nodes is nearly identical for both fairness notions. The 90-percentile wait time under BF is only about 1.9% smaller than that achieved under TF. Why is this? When there are multiple outstanding sessions with different speeds, the nodes with the slowest speed will, on average, complete last. Therefore, when more channel time is allocated to faster nodes, their wait time on average improves without significantly decreasing the wait time of slower sessions, which are highly likely to complete last under any work conserving policy. The phenomenon is more pronounced as the session size gets larger since a large session using the slowest speed is highly likely to complete last.

Figure 4-5 shows the distribution of wait time for all sessions. The figures do not give as much information as the previous figures since they include all the instances in which the wait time under each fairness notion is nearly the same. The median wait time under TF is 26.5% smaller than that under BF. However, the 90-percentile wait times under TF only improve over BF by 15.5%. This is because the slowest sessions with large amount of data to transfer will mostly be the ones that require a long time to complete and finish last under both fairness notions. Therefore, the impact of fairness notion on the expected wait time of all sessions at the tail of the distribution is less pronounced.

Figures 4-6(a) and 4-6(b) show CDFs of wait time similar to Figure 4-4(a) but for two different speed distributions, one in which most sessions (85%) use the fastest

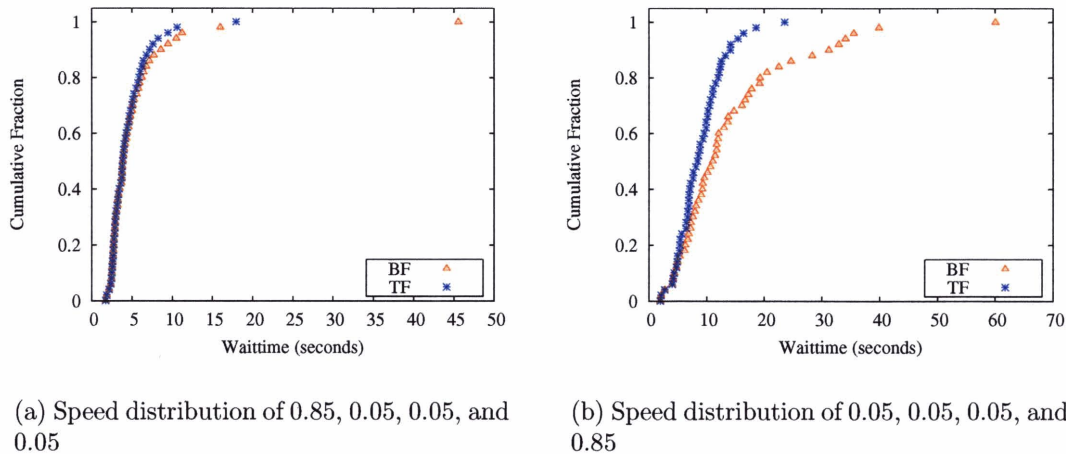


Figure 4-6: CDF of wait time of sessions as large as 1 MByte for two different distributions of 802.11b transmission speeds.

speed and the other in which most sessions use the slowest speed. As shown in the figures, the improvement in wait time is most pronounced when the traffic is largely dominated by nodes using slower speeds.

To summarize, favoring slower nodes in allocating channel time, as BF does, leads to significantly high average session wait time of faster nodes. Giving equal fractions of channel occupancy time to both slower and faster nodes reduce the expected session wait time of faster nodes with little impact on the average wait time of relatively lower nodes.

4.6 Impact of Greedy Channel Occupancy Time Allocations

The previous sections showed that compared to BF, TF leads to better performance in terms of aggregate throughput and average session wait time. This is because BF allocates lesser fractions of channel occupancy time to slower nodes. What happens if we allocate almost all of the channel occupancy time to the fastest node? In this section, we examine the performance impact of a greedy fairness notion (*GF*), under which the fair share of an entity transmitting at 11 is 0.99 and at 1 Mbps is 0.01. We compare GF against TF and BF.

Based on our throughput analysis (see Section 4.3), it should be clear that GF will achieve significantly higher aggregate throughput under our analysis model. The aggregate throughput under GF will approach what the faster node would achieve if it alone occupied the channel (since the slower node hardly get to transmit).

However, GF does not achieve similar improvements in session wait time. Figure 4-7(a) plots the aggregate expected wait time under all three fairness notions as a function of channel utilization, under the same model presented in Section 4.4. Al-

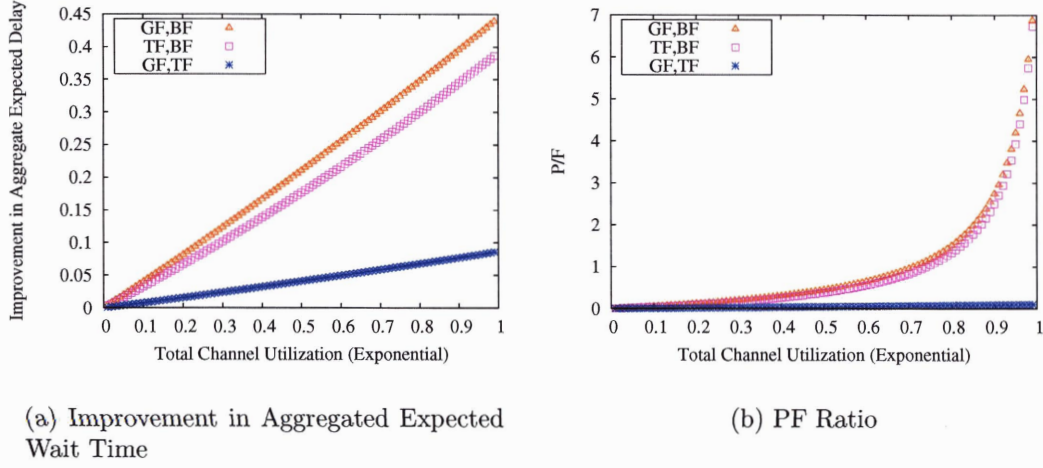


Figure 4-7: The improvement in aggregate expected wait time of all sessions achieved under TF over BF ($AggrDiff(TF, BF)$) and the PF ratio of three pairs of fairness notions.

though GF achieves improve aggregate expected wait time over the other two notions, the magnitude of improvement that it has over TF is relatively small (less than 9.5% in any case). Similarly, the PF ratio between GF and TF is a slowly increasing linear function of ρ as evident in Figure 4-7(b). Put it another way, GF does not yield much more additional utility improvement per amount of unfairness over TF than does TF over BF. This is because under GF, faster sessions achieve very high utility gain (i.e., reduction in delay). At the same time, slower sessions suffer significant utility loss as they get stuck behind faster sessions. Therefore, slower sessions that would otherwise have completed under TF, experience significantly higher delay under GF.

Finally, compared to TF, GF can lead to significantly larger relative worst-case wait time ratio in two-node competition:

$$\begin{aligned} WorstIndivRatio(GF, TF) &= \frac{0.50}{0.01} = 50 \\ WorstIndivRatio(TF, GF) &= \frac{0.99}{0.5} = 1.98 \end{aligned}$$

That is, the wait time of any individual session under TF cannot be larger than 1.98 times the wait time that it would observe under GF. However, the wait time of an individual session under GF can be as large as 50 times the wait time that it would achieve under TF.

4.7 Summary and Discussion

In this chapter, we developed measures to compare relative fairness and performance for two fairness notions. Using these measures, we evaluated various fairness notions including TF and BF, based on the analysis developed in the previous chapter as

well as through trace-driven simulation. In the presence of rate diversity in 802.11b networks, we showed that:

- In the context of two competing links, one exchanging data at 11 Mbps and the other at 1 Mbps,
 - The aggregate throughput under TF can be as much as 180% higher than that under BF, and
 - The $PF(TF, BF)$ of achieved throughput is about 6.2, showing that the achieved throughput of a slower node is not significantly smaller under TF than under BF, whereas the achieved throughput of a relatively faster node is much higher under TF than under BF.
- In the context of two classes of sessions, one exchanging data at 11 Mbps and the other at 1 Mbps, with an exponential arrival process and an exponential service time distribution,
 - The expected wait time under TF is 34% less than that under BF, when the channel utilization is 90%, and
 - The worst-case relative wait time ratio of TF to BF is about 1.8, showing that in the worst-case, a session under TF will take 80% longer to complete than it would under BF; however, the worst-case relative wait time ratio of BF to TF is 5.
- Using a trace-driven simulation in which the data rates used by competing sessions are uniformly distributed across four 802.11b data rates,
 - The median and 90-percentile wait time of all sessions under TF can be as much as 26.5% and 15.5% lower than that under BF, and
 - The median and 90-percentile wait time of sessions that are at least as large as 1 MByte can be as much as 8.3% and 42.4% lower under TF than under BF.

We also compared Greedy Fairness, which allocates a significantly higher fraction (99%) of channel occupancy time to faster nodes, against TF and BF. The results show that:

- GF can achieve higher aggregate throughput and lower expected delay than TF and BF.
- In the context of two classes of sessions, one exchanging data at 11 Mbps and the other at 1 Mbps, with an exponential arrival process and an exponential service time distribution,
 - The worst-case relative wait time ratio of TF to GF is about 1.98 whereas the same ratio of GF to TF is about 50, and

- The wait time improvement of GF over TF is relatively small (less than 9.5%), in the context of an exponential arrival process.

If achieving the highest possible aggregate utility is the only criteria, a MAC protocol should give large fractions of channel occupancy time to faster nodes (sessions) while starving slower nodes (sessions), under the following assumptions: i) when the nodes are always backlogged, ii) only UDP is used, and iii) the distribution of the amount of data to transfer does not depend on the data rate. Therefore, the aggregate UDP throughput and wait time under GF will be better than those under both TF and BF.

However, by starving slower nodes, GF can adversely affect TCP flows at slower nodes. Large per-packet delay can lead to adverse reactions from TCP, namely unnecessary TCP retransmissions because of timeouts and *duplicate-acks* etc. Unnecessary retransmissions lead to higher session wait time and wasted capacity.

Even without such implications, we show in the previous section that in the worst-case, GF can lead to significantly large wait time of sessions (much larger than they would under TF and BF) that use the lowest data rate.

Qualitatively, GF is not a “fair” notion if competing entities are considered to have equal priorities. On the other hand, TF provides equal divisions of the fundamental shared resource among competing entities, irrespective of the transmission method that they use, which is a reasonable and to us desirable fairness notion. TF significantly improves performance over traditionally accepted fairness notions like BF yet the worst-case relative performance such as the relative worst-case wait time ratio of TF to BF is relatively small.

Chapter 5

Distributed TES MAC Protocol

In this chapter, we describe TES (Time-fair, Efficient and Scalable MAC protocol), a distributed MAC protocol that:

- Achieves time-based, per-link fairness, and
- Provides long-term time-share guarantees among competing links,

These properties allow TES to:

- Improve throughput significantly relative to DCF in rate-diverse environments,
- Scale well, i.e., the aggregate throughput is sustained with increased load, and
- Improve the overall network efficiency without sacrificing fairness in the presence of channel errors, by encouraging nodes to employ link-layer burst loss avoidance schemes.

TES differs significantly from most existing MAC protocols in the way it adjusts the contention window of each competing node. Most MAC protocols adjust the *CW* of each transmitting node based on a feedback mechanism that informs (sometimes inaccurately) the transmitter whether its frame transmission is successful or failed because of collision. In contrast, TES sets the *CW* of each node based on its observed idle time preceding transmission events. Through an analysis in Section 5.2, we show that i) there exists an optimal amount of idle time preceding transmission events that maximizes aggregate throughput and ii) that optimal idle time varies little with the number of contenders. Each node running TES pre-computes the optimal idle time and adjust its *CW* dynamically so that its observed idle time equals the target idle time. Specifically, each node increases(decreases) its contention window size when its observed idle time preceding transmission events is smaller(larger) than the pre-computed target idle time. This mechanism allows TES to achieve and sustains a high collision-free channel utilization independent of the number of contenders.

Furthermore, unlike most MAC protocols, TES provides long-term fair channel occupancy time allocations among competing entities through a novel fairness controller. TES's fairness controller dynamically allocates TXOPs among contenders

based on their usages of channel occupancy time in the past. The next section defines terms that will be used throughout the chapter. Section 5.2 describes an analysis that leads to aforementioned key observations in achieving scalability. The rest of this chapter describes TES's operations in detail.

5.1 Definitions

Node *A* is within the *carrier-sense range* of node *B* if *A* can sense carrier most of the time when *B* transmits. Node *A* is within the *receive range* of node *B* if *A* can successfully decode the physical and MAC-layer headers of most of the frames transmitted by node *B*. In general, the carrier-sense range is at least as large as the receive range.

We define *contending* as the act of an entity competing for channel access against other entities, through a distributed channel access mechanism. We call nodes that are simultaneously contending *contenders*. In practice, the number of contenders is time-varying.

In the rest of this chapter, we assume that each node can transmit one frame within each TXOP. We postpone discussions on multiple frame transmissions in a burst until Section 5.7. A *transmission event* is a contiguous block of one or more frame transmissions that overlap. There are two types of transmission events: collision-free transmission events and collided transmission events. Each collision-free transmission involves exactly one frame transmission. A collided transmission event or *collision event* involves overlapping frame transmissions from two or more nodes. The duration of a collision event is the interval between the beginning of the earliest transmission and the end of the latest transmission involved in the collision. In Figure 5-1, there are a total of 3 frame transmissions but only 2 transmission events, one successful one and the other, a collision event.

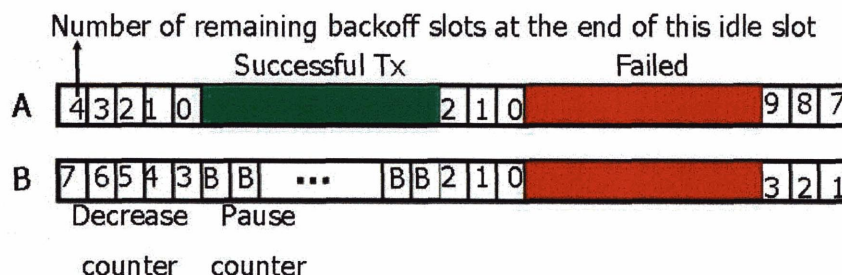


Figure 5-1: An example illustrating two different types of transmission events: a successful transmission event and a collision event. There are 3 frame transmissions but only 2 transmission events.

5.2 Analysis on Achieving Optimal Utilization

In this section, we conduct an average case analysis on achieving the optimal channel utilization when the underlying MAC protocol employs a contention window (CW) based channel access mechanism. We re-state the steps (mentioned in Section 2.1.8) that each node i takes in contending for a transmission opportunity using a CW based mechanism:

1. Sets a backoff counter to a random integer from a uniform distribution between 0 and CW_i
2. Decreases the counter for each time slot in which the channel is sensed idle,
3. Transmits a frame (a new frame or a retransmission) at the end of the idle slot when the backoff counter reaches 0,

DCF, TES, and many other protocols use this basic access method. Most protocols supplement it with stabilization techniques, such as exponential backoff, so that CW_i is dynamically adjusted based on observed load.

A major goal of a MAC protocol is to ensure that each contender i sets CW_i such that the collision-free channel utilization is maximized. The *collision-free channel utilization* is the fraction of channel occupancy time used for transmitting data frames without collisions. Achieving this goal requires balancing the average amount of idle time preceding transmission events and the collision rate, both of which are the causes of wasted capacity.

The analysis in this section is restricted to the case where each node i uses the same CW . We show a more general analysis in which nodes use different contention windows in Section B.1, which also validates our analytical results through simulation.

Let there be N_{cont} contenders. Let T_{idle} be the average amount of idle time preceding transmission events. Furthermore, let P_{col} be the collision probability, i.e., the expected ratio of the number of transmissions of each node i that are involved in collisions to the total number of transmissions of i . Again, since each contender uses the same CW , P_{col} is the same for all contenders in steady state. CW affects both T_{idle} and P_{col} . Specifically, the larger CW , the larger T_{idle} but the smaller P_{col} . Therefore, a MAC protocol must strike a balance between T_{idle} and P_{col} to maximize the collision-free channel utilization.

In this section, we show that:

- There exists a target collision probability, $TarP_{col}$, and the corresponding target idle time, $TarT_{idle}$, that maximize the collision-free channel utilization, independent of N_{cont} ,
- For a given number of contenders, N_{cont} , and a desired collision probability, $TarP_{col}$, we can compute CW so that the expected collision probability is close to $TarP_{col}$, and
- $TarT_{idle}$ can be pre-computed for a specific set of MAC-layer parameters.

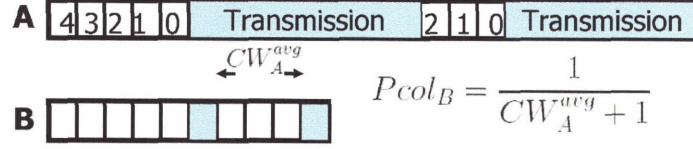


Figure 5-2: Node B observes node A 's frame transmission every CW_A^{avg} idle timeslots. Shaded slots represent the slots in which node B senses the busy channel.

5.2.1 Computing CW to Achieve a Desired Collision Rate

We start with a simple scenario in which node A and node B contend for channel access through a randomized CW -based backoff scheme as described earlier. Assume that the CW s of the two nodes are CW_A and CW_B respectively.

While listening to the channel, each node observes every transmission event. Figure 5-2 shows the idle-busy timeslot timeline of node B corresponding to node A 's transmission events. From node B 's perspective, on average it observes a transmission event (from another node) after every CW_A^{avg} idle timeslots. CW_A^{avg} is the average number of idle time slots preceding A 's transmission event. Since the backoff interval is randomly chosen over a uniform distribution,

$$CW_A^{avg} = \frac{CW_A}{2} \quad (5.1)$$

Node A 's transmission can be represented with a single busy slot on node B 's idle-busy timeline.

Since node B also selects a random number of backoff slots from a uniform distribution of 0 to CW , it has an equal probability to transmit at each timeslot in the idle-busy timeline. This is a correct assumption in all practical cases. However, there are corner cases where this assumption may not hold and we will discuss the limitations in Appendix B.1. Therefore, the probability of a node B 's transmission colliding with that of node A is:

$$P_{col_B} = \frac{1}{CW_A^{avg} + 1} \quad (5.2)$$

In other words, the probability of a node colliding with another node can be solely computed as a function of other nodes' average contention window size. This line of reasoning was proposed in [97]. We note that under their model, P_{col_B} is given as $\frac{1}{CW_A^{avg}}$, which is not accurate when CW_A^{avg} is very small, but has no significant differences with our P_{col_B} when CW_A^{avg} is reasonably large (which is the case in practice).

Our analysis focuses on the basic CW -based channel access mechanism. We derive $Tidle$ and establish relationships among $Tidle$, CW and $Ncont$. By extending Equation 5.2, we can compute the probability of contender i 's transmission not being involved in collisions, when there are $Ncont > 1$ contenders, each of which uses the

same CW and thus has the same average number of backoff time slots, CW^{avg} :

$$\begin{aligned} P_{nocol} &= \left(1 - \frac{1}{CW^{avg}+1}\right)^{N_{cont}-1} \\ \text{When } \frac{N_{cont}}{CW^{avg}+1} &\ll 1, \end{aligned} \quad (5.3)$$

$$P_{nocol} \approx e^{\frac{-(N_{cont}-1)}{CW^{avg}+1}}$$

The probability of contender i 's transmission colliding with another contender's transmission is:

$$P_{col} = 1 - P_{nocol} \quad (5.4)$$

Based on Equation 5.3, we can find out what CW^{avg} should be in order to achieve a particular collision rate P_{col} :

$$CW^{avg} \approx \frac{-(N_{cont}-1)}{\ln(1-P_{col})} - 1 \quad (5.5)$$

So, for each contender i to achieve the desired collision rate $TarP_{col}$, each must set their CW to $2 * CW^{avg}$, since CW^{avg} is the mean of a sequence between 1 and CW .

$$CW = 2 * CW^{avg} = 2 * \left(\frac{-(N_{cont}-1)}{\ln(1-TarP_{col})} - 1 \right) \quad (5.6)$$

Therefore, if each node i is aware of the number of contenders N_{cont} , it can set CW using the above equation so that $P_{col} \sim TarP_{col}$.

5.2.2 Relationships among $Tidle$, CW and N_{cont}

In this section, we analyze the relationships among $Tidle$, CW and N_{cont} . We first derive, $Ftxev$, the expected ratio of the number of transmission events to the number of transmissions (initiated by all contenders). When there are collisions, this ratio is less than 1.

Since each node observes the same P_{col} , the expected ratio of the number of frame transmissions involved in collisions to the total number of frame transmissions by all nodes, $P_{col_{all}}$ is also P_{col} . When each contender sets its CW using Equation 5.6, $P_{col} = TarP_{col}$.

Let $Ntxpercol$ be the average number of frame transmissions involved in each collision event. Note that $Ntxpercol \geq 2$. For each collision event, there are $Ntxpercol - 1$ additional frame transmissions, which should not be counted as transmission events. Therefore, the fraction of the transmissions that cannot be counted as transmission events is: $\frac{Ntxpercol-1}{Ntxpercol} * P_{col_{all}}$. Thus,

$$Ftxev = 1 - \left(\frac{Ntxpercol-1}{Ntxpercol} * P_{col_{all}} \right) \quad (5.7)$$

Based on $P_{col_{all}}$ and $Ftxev$, we can compute $P_{col_{ev_{all}}}$, the expected ratio of the

number of collision events to the total number of transmission events, as:

$$P_{col_{ev}}_{all} = \frac{\frac{P_{col_{all}}}{N_{txpercol}}}{F_{txev}} = \frac{P_{col_{all}}}{N_{txpercol} * F_{txev}} \quad (5.8)$$

We will now compute the average backoff delay, T_{idle} , preceding a transmission event. We provide our reasoning by examining the outcome of N_{cont} nodes competing from 0 to T seconds, where T is a reasonably long period. Let n_i denote the number of frame transmissions initiated by each contender i during T seconds. Since each node uses the same CW , at steady state, each node will be able to transmit the same number of frames. I.e., $n_i = n$. Let n_{alltx} and $n_{alltxev}$ be the total number of transmissions and transmission events respectively. Note that $n_{alltx} = \sum_{i=1}^{N_{cont}} n_i = n * N_{cont}$. Note also that $F_{txev} = \frac{n_{alltxev}}{n_{alltx}}$.

For each contender i , total amount of idle time, $T_{idle} * n_{alltxev}$, is $CW^{avg} * n * SlotTime$, where $SlotTime$ is the duration of each time slot in seconds. This is because, as shown in Figure 5-2, each idle slot on contender i 's idle-busy timeline is counted towards the backoff slots required for i 's transmissions. Dividing both sides with the total number of transmission events (observed by contender i), $T_{idle} = CW^{avg} * SlotTime * \frac{n}{n_{alltxev}}$. Since $n_{alltx} = n * N_{cont}$ and $F_{txev} = \frac{n_{alltx}}{n_{alltxev}}$, we have:

$$T_{idle} = \frac{CW^{avg} * SlotTime}{N_{cont} * F_{txev}} \quad (5.9)$$

Therefore, substituting Equations 5.6 and 5.7 in Equation 5.9 and using $N_{txpercol} = 2$,

$$\begin{aligned} T_{idle} &= \frac{\left(\frac{(-N_{cont}+1)}{\ln(1-TarPcol)} - 1\right) * SlotTime}{N_{cont} * \left(1 - \left(1 - \frac{1}{N_{txpercol}}\right) * TarPcol\right)} \\ &= \frac{(-N_{cont}+1 - \ln(1-TarPcol)) * SlotTime}{N_{cont} * \ln(1-TarPcol) * \left(1 - \frac{TarPcol}{2}\right)} \\ &= \frac{-N_{cont}+1 - \ln(1-TarPcol)}{N_{cont}} * \frac{SlotTime}{\ln(1-TarPcol) * \left(1 - \frac{TarPcol}{2}\right)} \end{aligned} \quad (5.10)$$

We will show in Section B.1 that for any reasonable collision rate (i.e., $P_{col_{all}} < 50\%$), $N_{txpercol}$ is very close to 2.

For N_{cont} larger than a dozen or so, T_{idle} is roughly constant:

$$\lim_{N_{cont} \rightarrow \infty} T_{idle} = \frac{-SlotTime}{\ln(1-TarPcol) * \left(1 - \frac{TarPcol}{2}\right)} \quad (5.11)$$

To sum up, the average amount of idle time preceding transmission event is roughly constant if the CW of each contender is set so that the expected collision rate is $TarPcol$.

5.2.3 Maximizing Collision-free Channel Utilization

We now derive the collision-free channel utilization as a function of $TarPcol$ and $Tidle$. Let $Tpayload$ be the average amount of channel occupancy time required to transmit payload. Let $Ttxev$ be the average amount of channel occupancy time required to transmit a frame. That is $Ttxev$ is the sum of $Tpayload$ and the overhead channel occupancy time required by the MAC and physical layers, namely the channel occupancy time necessary to transmit a physical layer preamble, the synchronous acknowledgment and the interframe space time between the data and ack frames and between data frames. Then, the aggregate collision-free channel utilization $Futil$ can be computed as:

$$\begin{aligned}
 Futil &= \frac{(1-P_{col_{all}}) * Tpayload}{Tpayload + T_{ovh} + Tidle} \\
 &\text{Substituting Equations 5.7, 5.8 and 5.11} \\
 &= \frac{2 * (1 - TarPcol) * Tpayload * \ln(1 - TarPcol)}{\ln(1 - TarPcol) * (2 - TarPcol) * Ttxev - 2 * SlotTime}
 \end{aligned} \tag{5.12}$$

We can then find $TarPcol$ that maximizes $Futil$.

$Tpayload$ depends on average frame size used and average data rate used. The overhead channel occupancy time largely depends on standard-specific parameters and $SlotTime$ is a standard-specific parameter. For example, for 802.11b devices that mainly transmit maximum-sized frames at 11 Mbps, $Tpayload = 1061.8 \mu s$ and $Ttxev = 1369.3 \mu s$ and $SlotTime = 20 \mu s$. $Futil$ reaches its maximum value of 0.6287 when $TarPcol = 0.1430$ ($P_{col_{all}} = 0.077$). For smaller values of $TarPcol$, the wasted capacity due to larger idle timeslots is more than the increased capacity due to reduced collision rate, thus reducing channel utilization. Similarly, larger values of $TarPcol$ degrade channel utilization since the wasted capacity due to collisions is more than the increased capacity due to reduced idle timeslots.

The optimal pre-computed $TarPcol$ can be used in computing CW for each contender so that $P_{col_{all}} \sim TarPcol$ (see Equation 5.6) if each contender knows $Ncont$. Estimating $Ncont$ is challenging for two main reasons: i) the number of contending nodes varies with time, and ii) the number of contenders can be drastically different from the number of nodes. The latter arises especially when the total load of the system is less than the capacity. In this case, the number of nodes competing for channel access at any moment will be less than the total number of nodes.

However, explicit estimation of $Ncont$ is not necessary. Based on our knowledge of ideal $TarPcol$, we can design a feedback based distributed protocol that will dynamically adjust CW_i of each node i according to observed $Pcol_i$. We will now use subscripts since both CW and $Pcol$ are time-varying and for any two contenders i and j , CW_i and $Pcol_i$ may not be identical to CW_j and $Pcol_j$ at any given point in time. However, an effective backoff mechanism will ensure that the steady state values of the two contenders are similar.

CW_i should be increased(decreased) if $Pcol_i$ is less(greater) than $TarPcol$. This approach, however, requires that each node i can observe $Pcol_i$ accurately. Unfortunately, unlike in wired networks, $Pcol_i$ is not easily observable in wireless networks since failed frame transmissions due to collisions are not easily distinguishable from

failed frame transmissions due to channel errors.

Unlike $Pcol_i$, $Tidle_i$ can be observed accurately. The optimal idle time, $TarTidle$, can be computed from $TarPcol$, independent of $Ncont$, as shown in Equation 5.11. We can then adjust CW_i based on pre-configured $TarIdle$ and observed $Tidle_i$. Specifically, CW_i should be increased(decreased) if $Tidle_i$ is less(greater) than $TarTidle$. When all nodes are listening to the channel all the time, $\forall i, Tidle_i \sim TarTidle$ in the steady state.

Our observation that $TarTidle$ can be obtained independent of $Ncont$ and thus can be used a reliable feedback for adjusting CW is the key in making our TES protocol more efficient than most existing protocols. The rest of this chapter describes TES's operations in detail.

5.3 Overview of TES

TES differs from existing MAC protocols in two major observable aspects.

High Utilization TES achieves and sustains a high collision-free channel utilization independent of the number of contenders through an effective link-layer collision control mechanism that is based on observed idle time, and

Fairness TES provides long-term fair channel occupancy time allocations through a fairness controller that dynamically allocates TXOPs among contenders based on their usages of channel occupancy time in the past.

In TES, both efficiency and fairness mechanisms are implemented in a *backoff instance* that is responsible for dictating when to transmit a frame. Each backoff instance is associated with a single outgoing link (defined by an AP-client pair). Since a client only associates with a single AP in typical 802.11 networks, TES running at the client will only have one backoff instance. However, the AP may have multiple backoff instances, one for each outgoing link to a client. Therefore, TES achieves per-link fairness by ensuring that each backoff instance will get an equal fraction of channel occupancy time in the long-term.

In a typical configuration, there will be a link-layer queue associated with each TES backoff instance. One attractive feature of TES is that it can achieve time-based, per-link fairness with a simple link layer queuing mechanism such as FIFO. Whenever there is a backlogged frame, the queuing mechanism will signal the associated TES backoff instance; the backoff instance will decide when to transmit that frame.

Like DCF, TES is a contention window based protocol. Under TES, each backoff instance i operates as follows:

1. Contends for a TXOP as follows:
 - (a) Sets a backoff counter to a random integer from a uniform distribution between 0 and CW_i ,
 - (b) Decreases the counter for each time slot in which the channel is sensed idle, and

- (c) Transmits a frame (a new frame or a retransmission) at the end of the idle slot when the backoff counter reaches 0, and
2. For every $TarNtxev$ transmission events being observed:
- (a) Computes CW_i so that the collision-free channel utilization is high, and
 - (b) Updates CW_i so that i achieves its fair time share in the long-term.

Unlike DCF and many other MAC protocols, TES does not adjust CW depending on the success or failure of the frame transmission. That is TES does not rely on a link-layer loss detection mechanism. Nor does TES rely on a particular method to avoid channel errors. Instead, TES provides flexibility in that it allows each node to: i) decide when to contend for channel access, i.e., when to request for a transmission opportunity, ii) what data rate and frame size to use, and iii) what link-layer retransmission mechanism to use. Nodes can make these decisions independently of TES, which dynamically adjusts CW_i to ensure that the collision-free channel utilization is high and that the node achieves its fair channel occupancy time in the long-term. Therefore, TES allows each node to implement time-varying transmission strategies without sacrificing fairness and the overall efficiency of the network.

When a node has more than one backoff instance (in the case of an AP with more than one link), multiple backoff counters may reach 0 in the same timeslot. In this case, TES randomly chooses a frame associated with a backoff counter to transmit. The other backoff counters are then set to random integers as described earlier.

```

PROCEDURE ENDTXEV() {
     $k$  : current round number
    Update  $Ntxev_\diamond^k$ ,  $Ttxev_\diamond^k$  and  $Tidle_\diamond^k$  based on the duration
        and amount of idle time preceding this transmission event

    if ( $Ntxev_\diamond^k \geq TarNtxev$ )
        RESETVARSIFFINACTIVE( $k$ )
        for each outgoing link  $i$  of this node
            UPDATEBACKOFFINSTANCE( $k, i$ )
        COMPUTEROUNDDURATION( $k$ )
        Increment  $k$ 
}

```

Figure 5-3: Pseudo-code of TES running at each node. Figure 5-4 describes remaining Procedures mentioned here.

Figure 5-4 presents the pseudocode of the round-based TES protocol. A round is a duration that spans $TarNtxev$ frame transmission events. For each frame transmission event observed, the procedure ENDTXEV is called and TES updates a set of variables that are relevant for the current, k^{th} , round. Then, for each backoff instance i , TES computes CW_i to achieve both fairness and efficiency through the procedure

(UPDATEBACKOFFINSTANCE). The next section describes what information is being monitored by each node running TES.

For the rest of this chapter, we use $Xabc_i^k$ to denote the variable $Xabc$ associated with link i during the k^{th} round; the X part identifies the type of the variable (e.g., integer, time, etc.) and the abc part describes it. When either the round number or link identifier is irrelevant, we replace it with \diamond . We use $\tilde{X}abc_i^k$ to denote an (moving) average value associated with link i at the k^{th} round.

5.4 Monitoring Transmission Events

At the end of each transmission event, the procedure ENDTXEV is called. Based on the duration of this transmission event and the amount of idle time preceding it, TES updates i) $Ntxev_\diamond^k$, the number of transmission events observed in this k^{th} round, ii) $Ttxev_\diamond^k$, the average amount of observed channel occupancy time per transmission event in this round, and iii) $Tidle_\diamond^k$, the average amount of observed idle time per transmission event in this round.

The duration of each transmission event and the amount of idle time preceding it can be easily obtained in any CSMA MAC protocol, though such information often unused by existing MAC protocols. As shown in Figure 5-1, each node listening to the channel will observe the alternating sequences of idle and busy time slots. In some cases, a node may not listen to the channel all the time. For instance, a battery-constrained node may turn off the radio when it is not expecting to exchange data with another node. We postpone our discussion on these issues until later in this chapter.

The end of the current round is reached when the number of transmission events observed reaches $TarNtxev$. TES then computes the values of variables that will be used for the next round using the information that it collected during the current round. For each outgoing link i , the contention window to be used for the next round, CW_i^{k+1} , is computed based on efficiency and fairness mechanisms as explained in the next two subsections.

5.5 Achieving High Collision-free Channel Utilization

TES relies on the analysis described in Section 5.2 to compute CW at the end of each round to achieve high collision-free channel utilization and scalability. As shown in Figure 5-4, procedure UPDATEBACKOFFINSTANCE, which is invoked for each outgoing link i at the end of round k , calls COMPUTECWFOREFFICIENCY to update CW_i . COMPUTECWFOREFFICIENCY computes CW_i^{k+1} based on the following observations made in Section 5.2:

- There exists an amount of idle time, $TarTidle$, that maximizes collision-free channel utilization and can be computed for a given set of MAC-layer parameters, and

- This ideal amount of idle time is roughly constant when $N_{cont} \geq 10$.

Specifically, CW_i^{k+1} is set to a value based on \tilde{CW}_i^k , the moving average of the contention windows that were previously computed by COMPUTECWFOREFFICIENCY. CW_i^{k+1} will be larger(smaller) than \tilde{CW}_i^k when $Tidle_o^k$ is smaller(greater) than $TarTidle$. We use the moving average of the contention window instead of the contention window computed at the end of the previous round, since doing so could lead to wider fluctuations in the computation of CW_i^{k+1} . CW_i^{k+1} will further be updated in the procedure UPDATECWFORFAIRNESS to ensure that each backoff instance achieves its long-term fair share of channel occupancy time. Notice that the moving average, \tilde{CW}_i^{k+1} , is updated before CW_i^{k+1} is adjusted by the procedure UPDATECWFORFAIRNESS. Doing so decouples the adjustment of CW for efficiency from the adjustment of CW for fairness.

The manner in which COMPUTECWFOREFFICIENCY computes CW_i^{k+1} for each backoff instance i ensures that:

- The observed amount of idle time preceding transmission events is similar to the ideal pre-computed value, $TarTidle$,
- Each (continuously) competing instance will converge to similar CW values, irrespective of their initial CW values, and
- The *settling time*, the time required for competing backoff instances to converge, is a small number of rounds. To converge is to reach the state at which the average of the CW values of each backoff instance is roughly the ideal CW .

We first focus on the core CW control mechanism of TES, which fulfills these properties in most cases, and then discuss an optimization that improves the settling time when the number of nodes is high and most nodes start with incorrect values of CW . We explain the challenges in achieving the aforementioned goals by using an example involving two competing nodes, i and j , that are continuously backlogged. Since both nodes are continuously monitoring the channel, the average amount of idle time observed by each node will approach the same value in a few rounds. Let's assume that this condition is met at the end of the k^{th} round. At this point, there are two distinct possibilities: $CW_i^k \sim CW_j^k$ or $CW_i^k > CW_j^k$.

In the first case, a sensible approach is to compute CW_i^{k+1} and CW_j^{k+1} by setting them to values larger or smaller than \tilde{CW}_i^k by multiplying or dividing by a constant factor, depending on whether the observed idle time is smaller or greater than the ideal amount. Such a mechanism will ensure that CW_i and CW_j will be increased and decreased in lock steps and that they will oscillate around the ideal CW value in future rounds.

Unfortunately, such a simple mechanism is not desirable when $CW_i^k > CW_j^k$, since doing so would constantly lead to $CW_i^{k+n} > CW_j^{k+n}$ for any $n \geq 1$, resulting in unfair allocations of transmission opportunities. To ensure convergence to the ideal CW value, the gap between CW_i and CW_j must be reduced over time. That

is if $CW_i^k > CW_j^k$, then in future rounds $n > k$, $\frac{CW_i^n}{CW_j^n} \leq \frac{CW_i^k}{CW_j^k}$. TES meets this requirement.

```

PROCEDURE UPDATEBACKOFFINSTANCE( $k, i$ ) {
    COMPUTECWFOREFFICIENCY( $k, i$ )
     $\tilde{CW}_i^{k+1} \leftarrow \text{EWMA}(CW_i^{k+1}, \tilde{CW}_i^k)$ 
    UPDATECWFORFAIRNESS( $k, i, CW_i^{k+1}$ )
}

PROCEDURE COMPUTECWFOREFFICIENCY( $k, i$ ) {
    if ( $TarTidle < Tidle_\diamond^k$ )
         $CW_i^{k+1} \leftarrow \frac{CW_i^k}{K_{base} + K_{dec}\sqrt{\tilde{CW}_i^k}}$ 
    else if ( $TarTidle > Tidle_\diamond^k$ )
        if ( $TarTidle < K_{diff} * Tidle_\diamond^k$ )
             $CW_i^{k+1} \leftarrow \tilde{CW}_i^k (K_{base} + \frac{K_{inc}}{\sqrt{\tilde{CW}_i^k}})$ 
        else
             $CW_i^{k+1} \leftarrow \tilde{CW}_i^k (K_{basehi} + \frac{K_{inc}}{\sqrt{\tilde{CW}_i^k}})$ 
    }

PROCEDURE EWMA( $val, avg$ ) {
    return  $\alpha * val + (1 - \alpha) * avg$ 
}

```

Figure 5-4: Pseudo-code of backoff instance i .

In the procedure COMPUTECWFOREFFICIENCY, when $TarIdle > Tidle_\diamond^k$, CW_i^{k+1} is computed in two ways depending on how large $TarIdle$ is compared to $Tidle_\diamond^k$. $K_{diff} \geq 1$ is a constant. The more complex way of increasing CW when $TarIdle > K_{diff} * Tidle_\diamond^k$ is an optimization to the core CW control mechanism, to improve settling time in cases when a large number of nodes start competing for channel access simultaneously, and will be explained shortly. The core CW control mechanism has two parts:

Decrease CW : Set \tilde{CW}_i^{k+1} by dividing \tilde{CW}_i^k with a factor, $(K_{base} + K_{dec}\sqrt{\tilde{CW}_i^k})$, where K_{base} and K_{dec} are constants, and

Increase CW : Set \tilde{CW}_i^{k+1} by multiplying \tilde{CW}_i^k with a factor, $(K_{base} + \frac{K_{inc}}{\sqrt{\tilde{CW}_i^k}})$, where K_{inc} is a constant.

We use the square root function instead of just \tilde{CW}_i^k to scale the large range of possible values that CW_i can take. For example, the ideal CW values for each backoff instance when there are 10 and 1000 contenders are about 102 and 11450 respectively. By using

the square root function, we scale CW_i in the range of 3.16 and 107 as opposed to the range of 100 to 11450 if the square root function were not used. A smaller range of multipliers or divisors is desirable since the sensitivity of the CW adjustment does not vary greatly with CW values (i.e., the number of contenders). In general, instead of the square root, we could have chosen any function of the form $\tilde{C}W^p$, where $p \leq 1$.

Both increase and decrease functions of TES meet the following requirements when $CW_i^k > CW_j^k$: $\frac{CW_i^{k+1}}{CW_j^{k+1}} < \frac{CW_i^k}{CW_j^k}$. In other words, each increase or decrease operation makes CW_i and CW_j closer to one another. The key here is that the scaling factors for both increasing and decreasing the CW vary depending on the value of the CW . Specifically, when increasing the CW , the *increase factor*, $(K_{base} + \frac{K_{inc}}{\sqrt{CW_i^k}})$, is less than $(K_{base} + \frac{K_{inc}}{\sqrt{CW_j^k}})$ whenever $CW_i^k > CW_j^k$. Similarly, the *decrease factor*, $\frac{1}{K_{base} + K_{dec}\sqrt{CW_i^k}}$, is greater than $\frac{1}{K_{base} + K_{dec}\sqrt{CW_j^k}}$ whenever $CW_i^k > CW_j^k$. As we show through simulation in Section 5.7, this mechanism for increasing and decreasing CW works well across a wide range of initial values.

Constant K_{base} is the minimum factor that multiplies or divides CW , independent of the value of the CW . We set $K_{base} > 1$ so that the CW at each round will be increased or decreased by at least a fixed amount, independent of the value of the CW . This is because in our core control mechanism, the increase factor decreases with the CW while the decrease factor increases with the CW . Therefore, at relatively small CW values, the increase factor can be much larger than the decrease factor, leading to a high settling time whenever there is an overshoot. Similarly, at high CW values, the increase factor can be much smaller than the decrease factor, leading to a high settling time whenever there is an undershoot. By setting $K_{base} > 1$, TES ensures that CW will be decreased and increased at the end of each round.

Constants K_{dec} and K_{inc} allow trade-offs between settling time and short-term unfairness. The larger these constants the faster it is for each node's average CW value approaches the ideal CW . However, larger values lead to larger fluctuations in the CW values of competing nodes. Wide fluctuations in CW values in turn lead to a higher degree of short-term unfairness since two nodes on short timescales can have CW values that are far apart (until they converge).

We now describe an optimization technique to improve the settling time in some cases. When a large number of nodes starts competing for channel access around the same time (e.g. large conferences where participants arrive at about the same time or in event-driven sensor networks [48]), the settling time under the core control mechanism may be relatively large, leading to low aggregate throughput (since the CW s of nodes are consistently smaller than the ideal value) and short-term unfairness among competing nodes. We optimize for this situation by speeding up the increase process of the CW when the observed amount of idle time is significantly smaller than the ideal amount of idle time. In this case, the increase factor is much larger than the one used in the core control mechanism, i.e., $K_{basehi} > K_{base}$. Note that $K_{diff} \geq 1$.

Why doesn't our CW increase/decrease mechanism adjust the CW as a function of the difference between the ideal and the observed idle time preceding transmission events? We choose not to do that because that will ultimately make CW_i too sensitive to its observed idle time, leading to unfair allocations in channel occupancy time among nodes that may persistently observe (slightly) different amounts of idle time. The amounts of idle time observed by competing nodes may vary slightly because of physical hardware implementations. For example, wireless cards with slightly different carrier sense thresholds may detect (slightly) varying amounts of idle time. Our algorithm is immune to such noisy samplings of idle time.

However, there are times when a node may consistently observe a drastically different amount of idle time than other competing nodes, e.g., in the presence of hidden terminals. When that happens the CW s, of nodes that observe drastically different amounts of idle time will be different. The hidden terminal problem is traditionally dealt with by using a floor acquisition mechanism such as the 802.11 RTS/CTS protocol to ensure that only a single node transmits within a radius that is twice larger than the receive range. When such protocols are used, each competing node within that radius will observe similar amounts of channel idle time.

Although our algorithm is shown in the context of 802.11-based networks, its scalability and robustness could be highly desirable beyond 802.11-based networks, such as sensor networks, where the number of nearby competing devices may number from a few dozens to a few hundreds. Unlike many existing MAC protocols [2, 63, 87, 102], TES achieves a bounded collision rate and a bounded amount of idle time per transmission event. This leads to sustained aggregate throughputs irrespective of the number of transmitters and their loads, as we show in Section 5.7.

5.6 Achieving Long-term Time-based Fairness

Upon computing CW_i as described in the previous section, each backoff instance i further updates CW_i so that it:

- Achieves its fair time share over a long period irrespective of its transmission strategy or that of any other nodes

TES's fairness control mechanism decouples selection of transmission strategy, i.e., when to contend for a transmission opportunity, what data rate and frame size to use and how many frames to transmit in a TXOP, from allocation of fair share of the channel occupancy time, i.e., how often to transmit. The former is the job of a separate link layer mechanism. The latter is carried out by TES based on the past usage of channel occupancy time by each link.

Each backoff instance i keeps track of its lead or lag of channel occupancy time above or below its fair share by maintaining $Tlag_i$ which reflects its cumulative lead or lag. A negative(positive) value of $Tlag_i$ denotes that link i has used more(less) than its fair share over the most recent active period of i . Each link i is considered *active* if it transmits or receives at least one frame transmission during $MaxInactTime$ seconds.

```

PROCEDURE UPDATECWFORFAIRNESS( $k, i, CW_i^{k+1}$ ) {
   $\tilde{Ttxev}_i^{k+1} \leftarrow \text{EWMA}(Ttxev_i^k, \tilde{Ttxev}_i^k)$ 
   $CW_i^{k+1} \leftarrow CW_i^{k+1} * \frac{\tilde{Ttxev}_i^{k+1}}{Ktxev}$ 

   $Tlag_i^{k+1} \leftarrow Tlag_i^k + (Ttottxev_\diamond^k * \phi_i - Ttxev_i^k)$ 
  UPDATETLAG( $k, i$ )

  if ( $Tlag_i^{k+1} > 0$ )
     $CW_i^{k+1} \leftarrow CW_i^{k+1} * (1 - LagMult * |Tlag_i^{k+1}|)$ 
  else if ( $Tlag_i^{k+1} < 0$ )
     $CW_i^{k+1} \leftarrow CW_i^{k+1} * (1 + LeadMult * |Tlag_i^{k+1}|)$ 

   $CW_i^{k+1} \leftarrow \min(\max(CW_i^{k+1}, MinCW), MaxCW)$ 
}

```

Figure 5-5: Pseudo-code of backoff instance i that provides long-term fair share guarantees. Figure 5-6 describes PROCEDURE UPDATETLAG.

Figure 5-5 shows the pseudo-code of TES's fairness mechanism. Procedure UPDATECWFORFAIRNESS updates CW_i^{k+1} . The main idea is to update CW_i^{k+1} in two steps:

1. Increase CW_i^{k+1} if the average duration of i 's transmissions is relatively large and decrease it otherwise.
2. Increase CW_i^{k+1} if i is leading in terms of achieved channel occupancy time, i.e., $Tlag_i^{k+1} < 0$, and decrease CW_i^{k+1} if i is lagging, and

The first step ensures that the share of transmission opportunities allocated is scaled according to the average duration of transmission so that each link achieves its fair share of channel occupancy time, irrespective of the average duration of its transmissions. The second step ensures that a link lagging behind another link will achieve a higher (expected) share of transmission opportunities in the future.

UPDATECWFORFAIRNESS scales CW_i^{k+1} according to the average duration of i 's transmissions so that a link, i , that has a smaller average duration of transmissions than another link, j , will be allocated a higher (expected) number of transmission opportunities in the ratio of the average channel occupancy time of i 's transmissions to constant $Ktxev$. We set $Ktxev$ to the expected amount of channel occupancy time required to transmit a maximum-sized frame at 11 Mbps. In general, each node observe similar values of the average channel occupancy time of all transmissions. Therefore, in steady state, for any two links i and j that have similar lags or leads:

$$\frac{CW_i^{k+1}}{CW_j^{k+1}} \sim \frac{\tilde{Ttxev}_j^{k+1}}{\tilde{Ttxev}_i^{k+1}}$$

However, each backoff instance may not be achieving its fair share of channel occupancy time for two main reasons: i) packets arrive in bursts, ii) the data rate and frame size used for each transmission may differ widely and on short timescales, and iii) nodes may delay transmissions to avoid time-correlated losses because of channel errors. In all cases, a backoff instance may have “missed” transmission opportunities, leading to unfair channel occupancy time allocation. TES attempts to provide long-term time share guarantees under those conditions by ensuring that a link that is lagging in channel occupancy time catches up with another link that is leading.

We now explain how TES adjusts the CW of each leading and lagging link. At the end of the k^{th} round, TES updates $Tlag_i^{k+1}$ by adding to it the difference between the target (or desired) amount of channel occupancy time of link i in the k^{th} round and the actual amount of channel occupancy time, $Ttxev_i^k$. The former is the product of the desired fair share of channel occupancy time, ϕ_i , and the total amount of channel occupancy time of transmission events during the k^{th} round, $Ttottxev_\diamond^k$.

If there are n active links with equal priorities, the fair share of each link i , ϕ_i , will be $\frac{1}{n}$. TES can also achieve weighted fairness, in which competing links have differing priorities and thus differing fair shares. The next subsection discusses how to configure ϕ_i in detail.

TES ensures that $Tlag_i^{k+1}$ is always bounded. That is $-MaxLagLead \leq Tlag_i^{k+1} \leq MaxLagLead$, where $MaxLagLead$ is a positive constant. This is achieved through Procedure UPDATETLAG, which we discuss in Section 5.6.2.

As shown in UPDATECWFORFAIRNESS, CW_i^{k+1} is increased with increase in $|Tlag_i^{k+1}|$. $LeadMult$ and $LagMult$ are positive constants. Notice that if $Tlag_i^{k+1}$ is unbounded, CW_i^{k+1} can potentially grow without bounds. This is undesirable since a link with a large $Tlag$ may have to wait for a long time before it can transmit. When most competing links have large $Tlag$, this leads to wasted capacity. When there are many long-lived links, all competing links could have large opponent leads for reasons explained in Section 5.6.2.

Finally, CW_i^{k+1} is bounded by $MinCW$ and $MaxCW$. $MinCW$ equals one-half of the ideal CW value that maximizes the collision-free channel utilization when there are two contenders. $MaxCW$ is the ideal CW value that maximizes the collision-free channel utilization when there are 2000 contenders.

To summarize, our method of adjusting CW_i^{k+1} allows each link to achieve its long-term fair channel time share, independently of i) the duration of its transmission and those of its competing links, and ii) how often a link contended the channel in the past.

5.6.1 Configuring a Desired Fair Channel Time Share

In the procedure UPDATECWFORFAIRNESS, the desired amount of channel occupancy time of link i in each round is computed according to its fair share, ϕ_i . In general, TES can achieve any value of ϕ_i , as long as $\sum_i \phi_i = 1$. In AP-based WLANs, ϕ_i of each active link (or client) i can be propagated by the AP to each of its clients. Each link i only needs to know its fair share and nothing else. This can be accom-

plished in two ways. First, the AP can explicitly inform each client of its fair share by i) including ϕ_i in the header of a frame transmission to client i , or ii) listing ϕ_i for all i in beacon control frames that the AP periodically transmit to advertise its data rate and other capabilities.

When there are $Nlink$ active links that have equal priorities, ϕ_i for each link i is $\frac{1}{Nlink}$. It is important to note that $Nlink$ is merely the number of active links, each of which may not be contending all the time. For instance if there are two clients, i and j , each with a low-bandwidth (UDP) audio streams and two clients, k and l , that are continuously sending UDP data packets, $Nlink$ is 4 and $\phi_i = \phi_j = \phi_k = \phi_l = \frac{1}{4}$. Over time, the two links with audio streams will be lagging and the two backlogged links will be leading. TES ensures that link i and link j will be lagging at the same rate. That is at steady state, $CW_i \sim CW_j$. Similarly, $CW_k \sim CW_l$. And, $CW_i, CW_j < CW_k, CW_l$ while the average idle time preceding transmissions is close to $TarTidle$.

5.6.2 Maintaining Fairness for Long-lived Active Links

The fairness mechanism described earlier provides time-based per-link fairness so long as $-MaxLagLead \leq Tlag_i^{k+1} \leq MaxLagLead$, i.e., the lead or lag accumulated so far has not reached its maximum value (in absolute terms). Unfortunately, this condition may not be met when competing links are active for a long time. When there are n competing links that are continuously backlogged, over time, for each link i , $Tlag_i^k < 0$, i.e. each link will think it's leading. This is because $Ttxev_i^k < \sum_i Ttxev_i^k$

in each round k in which there is any collision event, since two or more links involved in a collision event will be observing the transmission time of that collision event as its transmission time. Therefore, over time, $Tlag$ of each contending link will be negative and decreasing. This is not a problem if each competing link is leading by roughly the same amount all the time and $Tlag_i^k > -MaxLagLead$ ($|Tlag_i^k| < |MaxLagLead|$).

However, $Tlag_i^k$ will reach its minimum value, $-MaxLagLead$, if link i is active for a long enough interval. Resetting $Tlag_i^k$ to 0 may lead to unfair allocations of channel occupancy time among competing links. To see this, consider two uplink clients, link i and link j , that become active at the same time and are continuously backlogged. Because of collision events, in the long-term, both nodes will be leading, i.e., $Tlag_i^k < 0$ and $Tlag_j^k < 0$. Without loss of generality, assume that at the end of the k^{th} round $|Tlag_i^k| \geq MaxLagLead$ while $|Tlag_j^k| < MaxLagLead$, i.e., link i is leading more than link j . If $Tlag_i^k$ were reset to 0, it would appear that at the end of each future round, link j may be leading much more than link i and CW may be wrongly adjusted. For example, in the $k + 1^{th}$ round, since $|Tlag_i^{k+1}| < |Tlag_j^{k+1}|$, $CW_i^{k+1} > CW_j^{k+1}$. It is possible that in each round subsequent to the k^{th} round, link i will be allocated more and more channel time than link j , allowing $|Tlag_i^{>k}|$ to increase at a much more rapid pace than $|Tlag_j^{>k}|$. Therefore, in a future round m , $Tlag_i^m \sim Tlag_j^m$. By that time, link i has been allocated $|MaxLagLead|$ amount of channel time more than link j .

There are also scenarios where some long-lived links may be leading all the time

while others may be lagging all the time. One example is a TCP flow between two nodes. Assume that node i is sending TCP data packets to node j , which in turn send a TCP ack packet to node i for each TCP data packet received. Although the numbers of frame transmissions of the two nodes are similar, the total amount of channel occupancy time of node i 's transmissions is much higher than that of node j . Therefore, node i will always be leading and node j will always be lagging.

```

PROCEDURE UPDATETLAG( $k, i$ ) {
  if ( $Tlag_i^k < 0$  and  $|Tlag_i^k| \geq MaxLagLead$ )
    /*  $i$  has been leading for a while */
     $Rleadadv_{l \rightarrow k, i} \leftarrow \frac{|Tlag_{l \rightarrow k, i}|}{Ttxev_{l \rightarrow k, \diamond}}$ 
     $\tilde{Rleadadv}_i^{k+1} \leftarrow \text{EWMA}(Rleadadv_{l \rightarrow k, i}, \tilde{Rleadadv}_{l, i})$ 
     $Tlag_i^{k+1} \leftarrow Tlag_i^k - \frac{MaxLagLead}{2}(1 - \tilde{Rleadadv}_i^{k+1})$ 
     $Tlag_i^{k+1} \leftarrow \max(Tlag_i^{k+1}, MaxLagLead)$ 
     $l \leftarrow k + 1$ 

  else if ( $Tlag_i^k > 0$  and  $Tlag_i^k \geq MaxLagLead$ )
    /*  $i$  has been lagging for a while */
     $Rlagadv_{l \rightarrow k, i} \leftarrow \frac{|Tlag_{l \rightarrow k, i}|}{Ttxev_{l \rightarrow k, \diamond}}$ 
     $\tilde{Rlagadv}_i^{k+1} \leftarrow \text{EWMA}(Rlagadv_{l \rightarrow k, i}, \tilde{Rlagadv}_{l, i})$ 
     $Tlag_i^{k+1} \leftarrow Tlag_i^k - \frac{MaxLagLead}{2}(1 - \tilde{Rlagadv}_i^{k+1})$ 
     $Tlag_i^{k+1} \leftarrow \min(Tlag_i^{k+1}, MaxLagLead)$ 
     $l \leftarrow k + 1$ 

  /*  $MaxLagLead \leq Tlag_i^{k+1} \leq MaxLagLead$  */
}

```

Figure 5-6: Pseudo-code of backoff instance i that concerns with fairness.

TES:

- Provides fair allocations of channel occupancy time among all leading links (that are backlogged), and
- Provides fair allocations of channel occupancy time among all lagging links (that are backlogged).

Procedure UPDATECLAGLEAD in Figure 5-6 is critical in achieving these properties for long-lived links. The main idea behind UPDATECLAGLEAD is to reduce $Tlag_i^{k+1}$ by an amount, called *lead or lag advance*, that depends on the rate of lead or lag advance of link i . The *rate of lead or lag advance* of link i , $\frac{Tlag_i}{Ttxev}$, is the absolute change in lead or lag accumulated by link i over the total amount of time required to transmit frames. Specifically, $Tlag_i^{k+1}$ is updated whenever it reaches the maximum value in such a way that

- Each leading link achieves an equal rate of advance in lead, and

K_{inc}	0.6
K_{dec}	0.0075
K_{base}	1.01
K_{base2}	1.75
K_{diff}	4.5
$TarNtxev$	5
$LeadMult$	0.75
$MaxLagLead$	0.3 second
$LagMult$	$\frac{1}{1-LeadMult} = 4.0$

Table 5.1: Configurable TES parameters and their default settings.

- Each lagging link achieves an equal rate of advance in lag,

When $Tlag_i^{k+1} < 0$ and $|Tlag_i^{k+1}| \geq MaxLagLead$, the procedure UPDATECLAGLEAD first computes $Rleadadv_{l \rightarrow k, i}$, the (absolute) rate of lead advance of link i since the last time $Tlag$ was updated at the end of the $l - 1^{th}$ round. Based on $Rleadadv_{l \rightarrow k, i}$, its moving average, $\tilde{Rleadadv}_i^{k+1}$, is updated. UPDATECLAGLEAD then updates $Tlag_i^{k+1}$ by subtracting by an amount that depends on $\tilde{Rleadadv}_i^{k+1}$ so that $Tlag_i^{k+1} \leq MaxLagLead$. $Tlag_i^{k+1}$ decreases with increase in $\tilde{Rleadadv}_i^{k+1}$. When $\tilde{Rleadadv}_i^{k+1}$ reaches its maximum possible value of 1, $Tlag_i^{k+1} = \frac{MaxLagLead}{2}$. Therefore, the larger the rate of lead advance of link i , the larger $Tlag$. Recall that in the procedure UPDATECLAGLEAD, the more negative $Tlag$ is, the larger CW is. Thus, a link with a higher lead advance rate will be allocated a relatively lower amount of channel occupancy time (through a relatively larger CW) than the link with a lower lead advance rate.

Similarly, when link i is lagging for a long time and $Tlag_i^{k+1} \geq MaxLagLead$, UPDATECLAGLEAD computes $Rlagadv_{l \rightarrow k, i}$, the average rate of lag advance from round l through round k , and reduce $Tlag_i^{k+1}$ proportional to $(1 - Rleadadv_{l \rightarrow k, i})$. Therefore, a link with a higher lag advance rate will be allocated a relatively higher amount of channel occupancy time than the link with a lower lag advance rate.

5.7 Evaluation

In this section, we evaluate TES using the *ns-2* simulator [76]. We compare the performance of TES against DCF, the standard 802.11 MAC protocol, and IDS (for IdleSense), a recently proposed idle time based channel access protocol [40]. Each data point is an average of ten runs of simulation. Unless otherwise noted, each simulation run lasted for about 100 seconds and competing nodes are 802.11b-based.

We compute the pair of collision rate and idle time that maximizes the aggregate throughput when frames are transmitted at 11 Mbps based on the analysis in Section 5.2.3. We find that the target fraction of collision events relative to the total number of transmission events ($P_{coleval}$) should be about 8.7%. The corresponding

ideal amount of idle time is about $125 \mu s$. DCF requires that each frame transmission be preceded by an idle interval of $50 \mu s$ in addition to a randomized backoff delay. For a fair comparison among all three protocols, each node under both TES and IDS only initiates backoff after a $50 \mu s$ of idle period. Both IDS and TES use the same target idle time.

For IDS, we use the default parameters for adjusting CW as suggested in [40]. Table 5.1 lists the default parameters of TES. We achieve these by conducting several simulations.

Under DCF, the CW is lower-bounded at 31 and upper-bounded at 1024 as specified in the 802.11 specification [2]. IDS did not propose that the CW is upper-bounded [40]. Although we propose that the CW is upper-bounded under TES, for a fair comparison between IDS and TES, we ran both TES and IDS without upper-bounding the CW . The CW is also lower-bounded at 6 under both TES and IDS.

We first examine the benefits of TES without the fairness mechanism described in Section 5.6. We evaluate TES with its fairness mechanism in Section 5.7.6 and beyond.

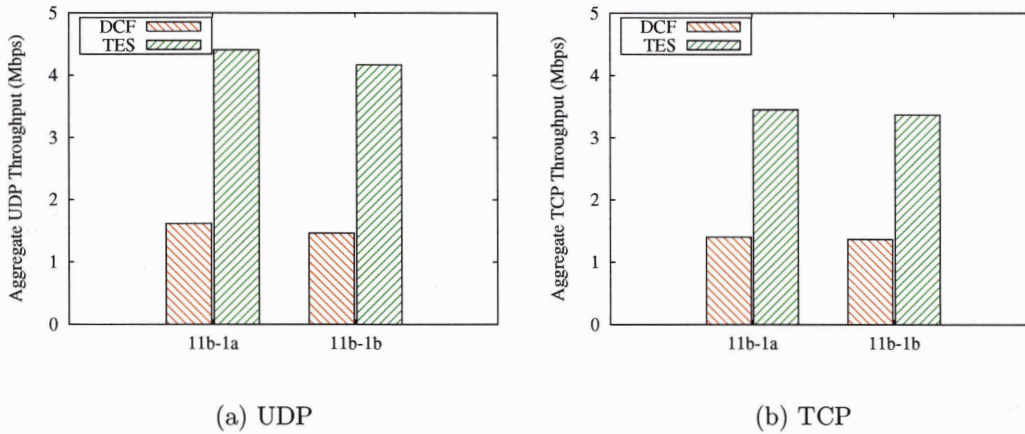


Figure 5-7: Aggregate UDP and TCP throughputs achieved in either the uplink direction (11b-1a) or downlink direction (11b-1b) by four competing 802.11b clients, two sending at 11 Mbps and the other two at 1 Mbps.

5.7.1 Benefits of Time-based Fairness

We ran experiments to show that TES achieves time-based fairness, leading to increased aggregate throughput relative to DCF in rate diverse environments. Figure 5-7 shows the aggregate UDP and TCP throughputs achieved by two nodes exchanging data with the AP at 11 Mbps and two other exchanging data with the AP at 1 Mbps. In 11b-1a, all flows are in the downstream direction and in 11b-1b, all are in the upstream direction.

TES achieves time-based fairness and as a result, the aggregate UDP throughput in each scenario improves by 173%. Under both schemes, the channel occupancy time

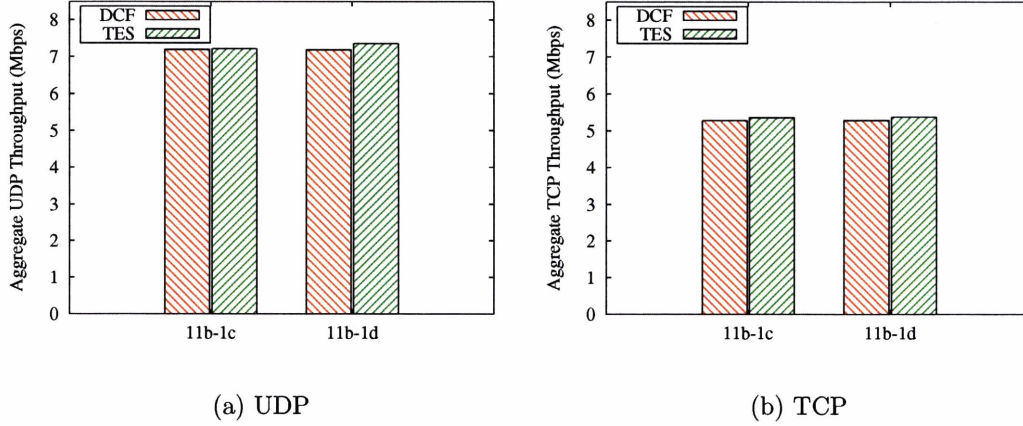


Figure 5-8: Aggregate UDP and TCP throughputs achieved by four competing 802.11b clients, each exchanging data at 11 Mbps. In 11b-1c, there are 2 uplink and 2 downlink clients. In 11b-1d, there are 1 uplink client and 3 downlink clients. Although the aggregate throughput is roughly the same for both scenarios, fairness is affected (see Figure 5-9)

achieved by each link is no more than 2% of that of another link (not shown in the figure).

IDS can also achieve time-based fairness for these scenarios. However, IDS, as proposed in [40], will not achieve time-based fairness in many practical scenarios. To achieve time-based fairness, IDS attempts to scale transmission probability based on the data transmission rate. This leads to time-based fairness if the data transmission rate is the only factor that can influence the transmission time of each frame, which is not typically the case. For example, if the transmission probability of each node is only scaled according to its data rate, competing nodes using different frame sizes but the same data rate will not achieve equal amounts of channel occupancy time.

5.7.2 Benefits of Per-link Fairness

In this section, we show that TES provides per-link fairness. Figure 5-8 shows the aggregate UDP and TCP throughputs achieved by four competing 802.11b clients in two scenarios. All clients exchange data with the AP at 11 Mbps. In 11b-1c, there are 2 upstream clients and 2 downstream clients whereas in 11b-1d, there are 3 downstream clients and only 1 upstream client. As expected, the achieved aggregate throughputs under two different schemes are very similar for each scenario (since all nodes use the same data rate).

However, the amount of channel occupancy time allocated to each link is different. Figure 5-9(a) shows the ratio of the maximum amount of channel occupancy time achieved by a link to the minimum amount achieved by a link. All flows are UDP flows. In scenario 11b-1c, the max/min ratio of channel occupancy time is roughly 1

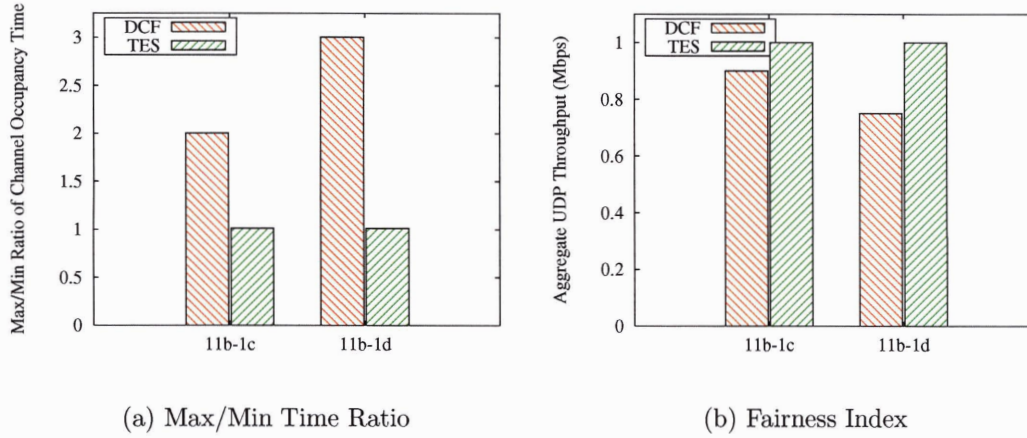


Figure 5-9: Max/min ratio and time fairness index of two scenarios with UDP-only flows. In each scenario, TES achieves a very high degree of fairness, with both the max/min ratio and fairness index close to 1. DCF allocates less channel occupancy time to downstream links, leading to much higher degrees of unfairness among competing links.

under TES but 2 under DCF. This is because the two downstream links are allocated the same amount of channel occupancy time as each upstream link. Similarly, in scenario 11b-1d, the max/min ratio of channel occupancy time under DCF is 3.

Figure 5-9(b) shows the fairness index [46] of channel occupancy time allocated to links. A fairness index of 1 indicates that each link achieves an equal amount of channel occupancy time. Again, TES achieve fairness induces close to 1 for each scenario. The differences in fairness index under DCF and TEA are not as dramatic as the max/min ratios because the fairness index computes the “average” deviation of each node’s allocated share of resource from its fair share.

5.7.3 Variations in Frame Size

Figure 5-10 shows the aggregate UDP throughput and max/min ratio of channel occupancy time under each of the three MAC protocols, when two nodes use the maximum-sized frames and the other two use a payload size 50% smaller than the maximum size allowed. The aggregate throughput is roughly equal. However, neither IDS nor DCF provide time-based fairness since nodes using the larger payload size get a higher share of channel occupancy time than those using the smaller payload size. In contrast, TES achieves time-based fairness, irrespective of the frame size. Note that the max/min time ratio under both IDS and DCF is about 1.67 and not 2. This is because of a constant per-frame MAC and physical layer overhead independent of the payload size.

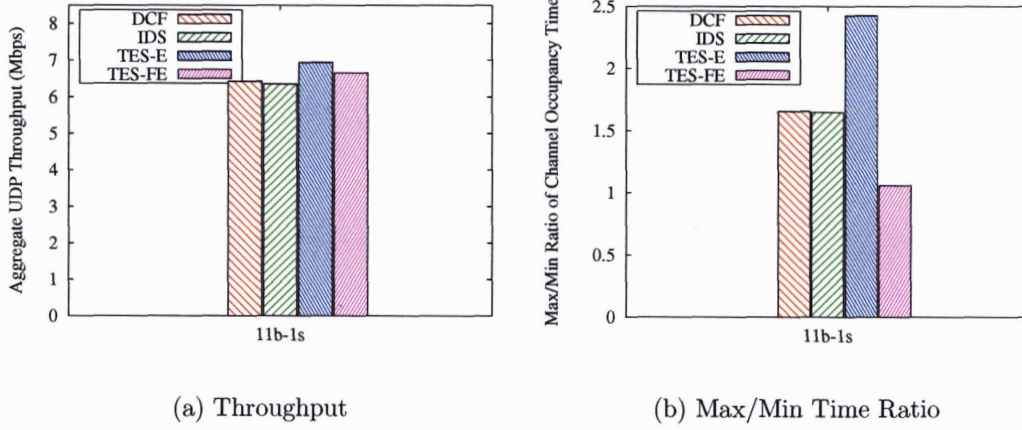


Figure 5-10: Aggregate UDP and max/min time ratios when there are four competing 802.11b clients, with two using a payload size twice that of the other two.

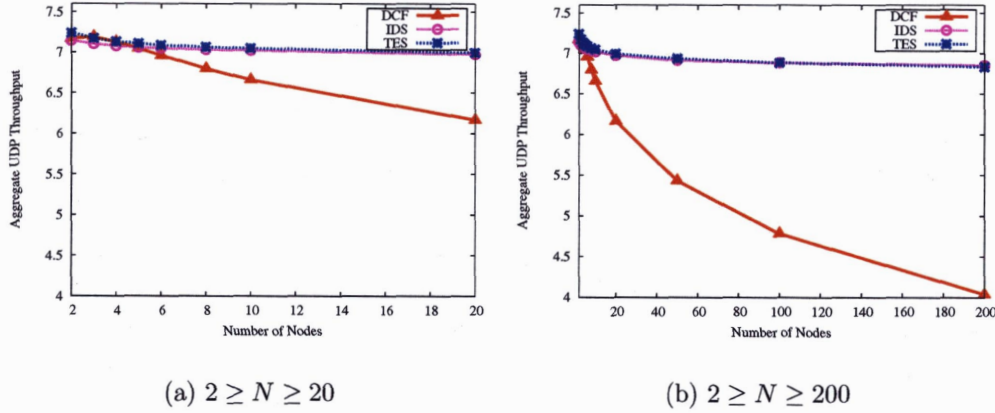
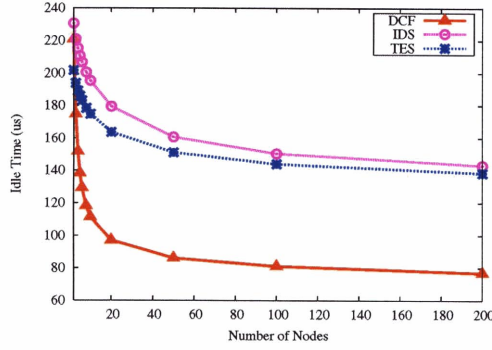


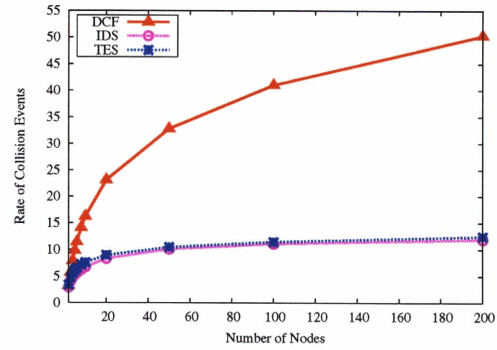
Figure 5-11: Aggregate UDP Throughput as a Function of the Number of Contenders.

5.7.4 Achieving Scalability

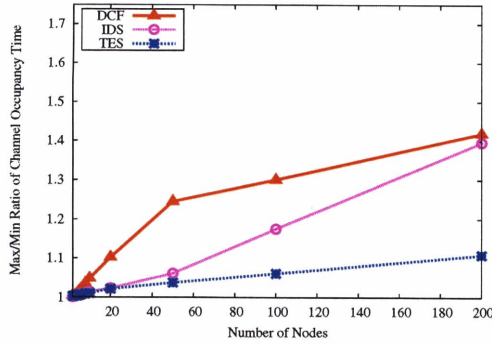
In this section, we show how TES sustains aggregate throughput independent of load. We ran a number of experiments involving various numbers of continuously backlogged nodes. Figures 5-11(a) and 5-11(b) plots the aggregate UDP throughput achieved under each scheme as a function of the number of backlogged nodes, which are all sending UDP data packets to a common AP using a data rate of 11 Mbps. Each simulation run lasted about 300 seconds. The aggregate throughput for all three schemes, DCF, IDS and TES, are comparable when the number of nodes is small. When the number of nodes is 10, TES achieves 5.8% higher aggregate throughput than DCF. The throughput gain increases to 13.5%, 28.3% and 71%, when the number of nodes is 20, 50 and 200 respectively. TES achieves slightly higher aggregate UDP throughput than IDS in each case.



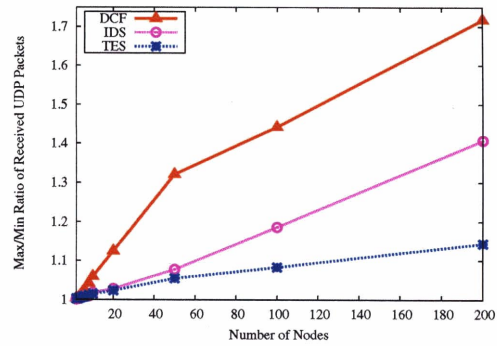
(a) Average Amount of Idle Time Preceding Transmission Events



(b) Rate of Collision Events



(c) Max/Min Ratio of Channel Occupancy Time



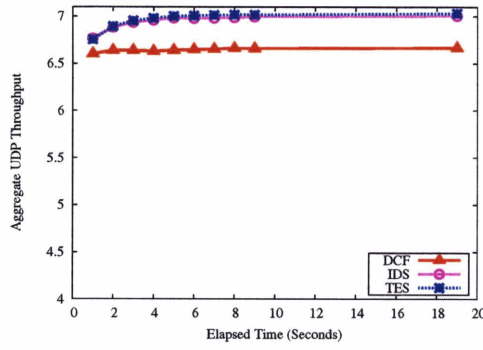
(d) Max/Min Ratio of Received UDP Packets

Figure 5-12: Measures related to efficiency and fairness when different numbers of backlogged nodes are sending UDP data packets to a common AP.

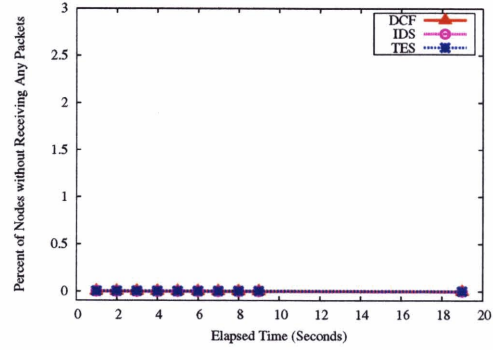
The performance gain achieved by TES and IDS over DCF can be explained by Figure 5-12(a), which plots the average amount of observed idle time preceding transmission events, and Figure 5-12(b), which plots the ratios of collision events to transmission events. The amount of idle time that maximizes the aggregate throughput according to the analysis in Section 5.2.3 is about $175\mu s$. As shown in Figure 5-12(a), the observed idle time under TES tracks most closely to this number and as a result the aggregate throughput under TES is superior to the other two schemes.

TES also achieves a much higher degree of fairness than both DCF and IDS. Figure 5-12(c) compares the max/min ratios of channel occupancy time achieved under all three schemes. TES achieves max/min ratios close to 1. Similarly TES achieves better max/min ratios of received UDP packets than the other two schemes, as shown in Figure 5-12(d).

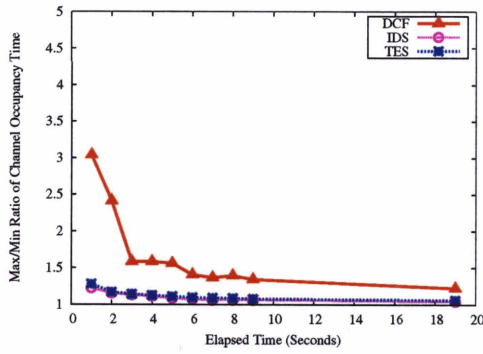
To summarize:



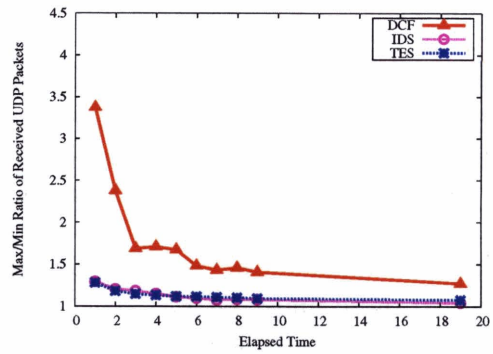
(a) Aggregate Throughput



(b) Percentage of Nodes without Any Successfully Transmitted Packets



(c) Max/Min Ratio of Channel Occupancy Time



(d) Max/Min Ratio of Received UDP Packets

Figure 5-13: Evolutions of various measures over the first 19 second since 10 nodes simultaneously computed for channel access.

- The aggregate throughput under DCF:
 - Is significantly less than those under both IDS and TES when the number of competing nodes is more than a dozen, and
 - Decreases rapidly with the number of contenders whereas those under IDS and TES remain relatively unchanged with increased number of contenders, and
- TES is much fairer than DCF in all the cases examined and than IDS when the number of nodes is large. The relative improvement in fairness that TES has over both DCF and IDS increases with the number of nodes.

5.7.5 Convergence To Fairness

In this section, we explore how quickly each scheme reaches i) the state at which the amount of observed idle time approaches the ideal amount and remains that way, and ii) the state at which the *CW*s of the competing nodes are similar and oscillate around the ideal value. The experiments were similar to the scalability experiments except that they were run at much shorter time scales. The four sub-figures of Figure 5-13 plot the evolutions of various measures achieved under the three schemes when there are 10 contenders sending data at the same time at a data rate of 11 Mbps. As shown in Figure 5-13(a), it took about 2 to 3 seconds for TES and DCF to reach the state at which the aggregate throughput remains roughly unchanged with further increase in elapsed time. IDS took about 5 seconds to reach that state. Figure 5-13(b) plots the percentage of nodes, that could not successfully transfer even a single UDP packet to the AP. Even within the first second, the percentage was zero under all three schemes, i.e., all nodes were able to successfully transfer at least one UDP packet to the AP.

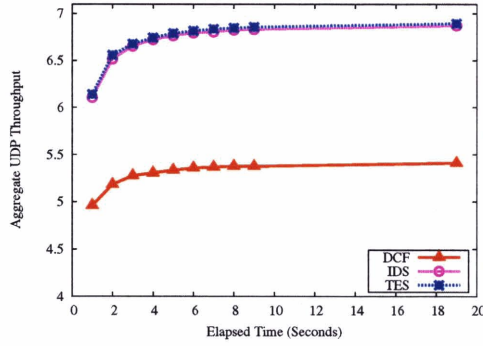
The max/min ratios of channel occupancy time and received packets at smaller amounts of elapsed time were significantly higher under DCF than under the other two schemes. This is because the *CW*s of the competing nodes often vary widely at the beginning. Under the exponential backoff scheme used by DCF, a few unfortunate nodes that experience successive losses because of collisions will have relatively large *CW*s. Since both IDS and TES do not react to frame losses, they are immune to fluctuations in frame loss rates. Hence, the max/min ratios under IDS and TES are superior to those under DCF.

DCF's inability to quickly converge to the state at which each competing node achieves roughly an equal amount of channel occupancy time, is more pronounced with increased number of nodes. As shown in Figure 5-14(c), when there are 50 contenders, the max/min ratio of channel occupancy time under DCF is as high as 9 whereas those of IDS and TES remain below 3. Furthermore, as shown in Figure 5-14(b), the AP did not receive any packets from 7.5% of nodes during the first second.

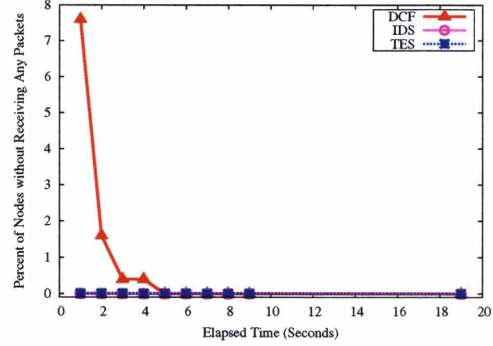
TES consistently achieves lower max/min ratios than IDS. The advantage of TES over IDS becomes more pronounced with increased number of nodes. Figures 5-15 and 5-16 plot the same sets of measures for TES and IDS when the number of nodes is 100 and 200 respectively. We omit DCF from these figures to highlight the differences between IDS and TES. The measures under DCF are worse than those under both IDS and TES for both 100-node and 200-node competitions.

As shown in Figure 5-15, when 100 nodes compete for channel access, TES is consistently better than IDS under all four measures. The differences in the max/min ratios are most pronounced.

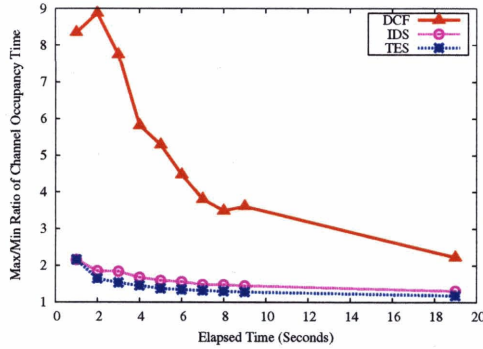
When 200 nodes compete for channel access, the improvements that TES has over IDS are drastic especially in fairness related measures. As shown in Figure 5-16(b), there were significantly smaller fractions of nodes, from which the AP did not successfully receive any UDP packet, under TES than under IDS, for the first 19 seconds. Figure 5-16(c) shows that the max/min ratio of channel occupancy time under IDS was still diverging even after 19 seconds had elapsed. In contrast, the max/min ratio of channel occupancy time rapidly converges under TES. Notice that



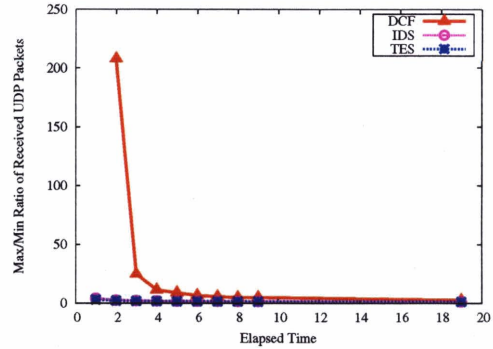
(a) Aggregate Throughput



(b) Percentage of Nodes without Any Successfully Transmitted Packets



(c) Max/Min Ratio of Channel Occupancy Time



(d) Max/Min Ratio of Received UDP Packets

Figure 5-14: Evolutions of various measures over the first 19 second since 50 nodes simultaneously competed for channel access.

during the first three seconds, the max/min ratios under TES are higher than those under IDS. This is because during the first few seconds, each node under TES rapidly increases CW according to its observed amount of idle time which was constantly smaller than the ideal amount at the end of early rounds.

The max/min ratio increases with increased CW . However, this process of continued increase in CW reverses quickly under TES as the observed amount of idle time under each node becomes greater than or equal to the ideal amount of idle time. This condition is reached in just 2 seconds. At that point, each node under TES begins to reduce its CW and continues the increase/decrease process, thereby lowering the max/min ratio. On the other hand, it takes IDS a relatively long time to reach the point where the average amounts of idle time observed by nodes reach or exceed the ideal amount of idle time. The significant gains achieved by TES over IDS are mainly attributable to TES's method of adjusting CW as a function of the difference between the ideal and observed amounts of idle time.

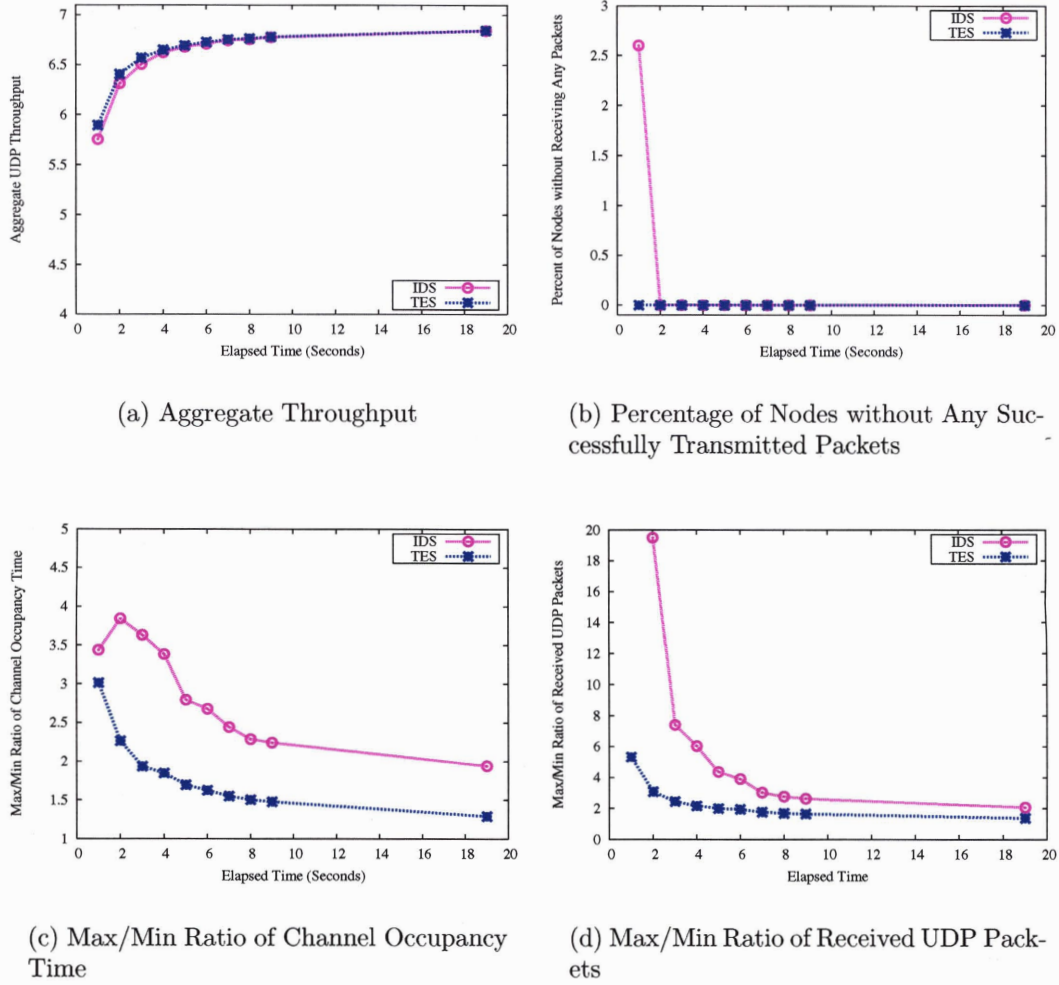


Figure 5-15: Evolutions of various measures over the first 19 second since 100 nodes simultaneously competed for channel access.

Figure 5-17 plots Jain's fairness index of received UDP packets of all three schemes. Jain's fairness index [46] quantifies the aggregate variations between an actual weight vector and the desired fair weight vector. In the aforementioned experiments, each node should have been able to successfully transmit roughly the same number of UDP packets. I.e., the fair weight of each node will be the average number of received UDP packets. The fairness index can vary between 0 and 1, with 1 indicating that the two vectors are identical. As shown in the figure, TES achieves a higher or equal degree of fairness than both IDS and DCF in all cases.

To summarize:

- Both TES and IDS achieve higher aggregate throughput and fairness than DCF,
- TES achieves noticeable gains in aggregate throughput over IDS during the first few seconds after nodes began competing for channel access and

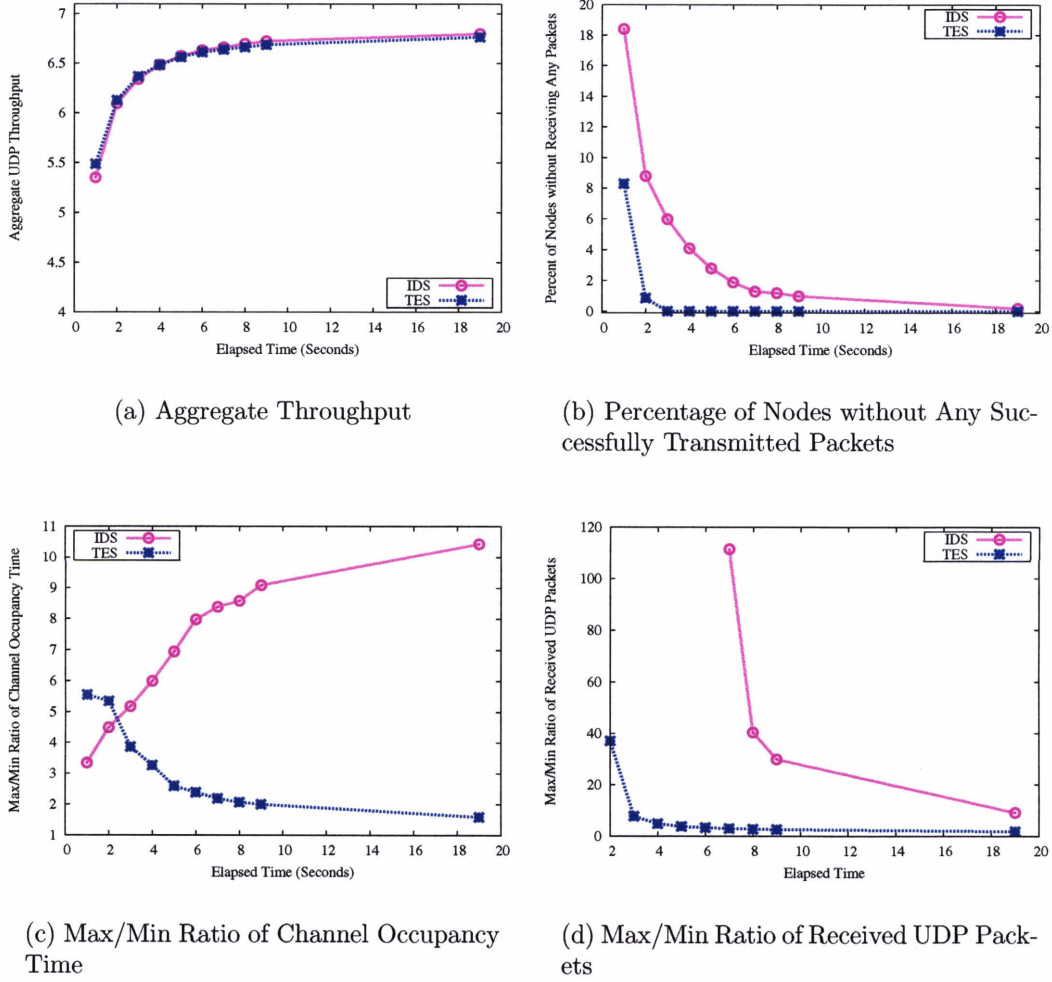


Figure 5-16: Evolutions of various measures over the first 19 second since 200 nodes simultaneously competed for channel access.

- TES achieves significantly higher degrees of fairness than IDS during the first several seconds after nodes began competing for channel access, whenever the number of contenders is more than a few dozens.

5.7.6 Random Channel Losses

We now explore how various MAC protocols react to two types of losses due to channel errors: random losses and burst losses. In some environments, channel losses appear random, whereas in other environments, especially mobile environments, channel losses tend to occur in bursts [73, 71, 83]. This subsection explores the impact of random losses in detail and the next subsection examines the impact of burst losses.

The key question in both cases is what to do when losses occur. If the channel losses experienced by a node are random, the node is not likely to benefit from delaying

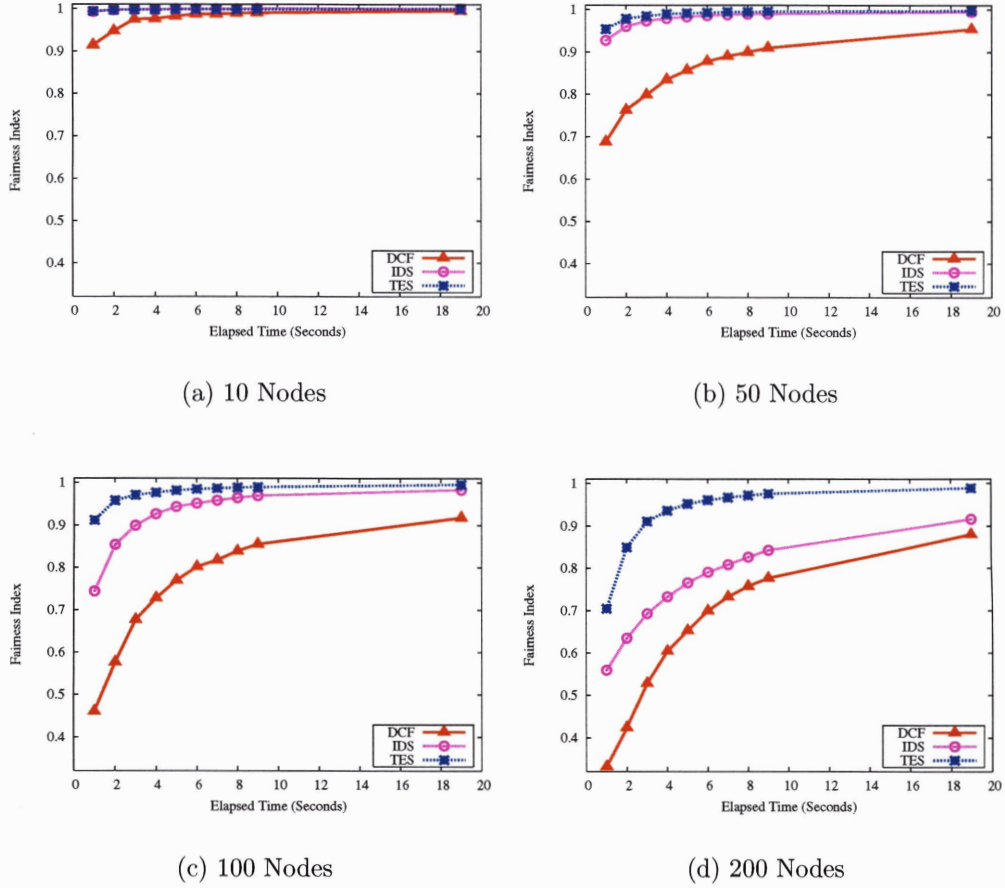


Figure 5-17: Jain's Fairness Index of Received UDP Packets.

its next transmission. However, if the losses occur in bursts, the node should delay its transmissions and let other nodes transmit instead. In practice, a node may not know for sure which type of errors it is experiencing. The type of channel errors may vary over time depending on the location and the movement of the mobile node. Therefore, it is important that any scheme that attempts to avoid burst losses must not adversely affect the performance and fairness in the presence of randomized errors. This subsection and the next explore four different schemes: DCF, IDS, TES-E and TES-FE.

TES-E is TES with a simple exponential scheme to avoid burst losses, which works as follows. Whenever a node encounters a loss and the average fraction of channel occupancy time used by the node is less than 75% (i.e., there is active competition for channel access), it will add to the normal randomized backoff delay a fixed amount of time. This amount is doubled with each successive loss. This exponential scheme is similar but not identical to the one used by DCF. TES-FE is TES with the same exponential scheme and with the fairness mechanism described in Section 5.6. As shown in this section and the next, TES-FE is the only one of the protocols that can

achieve high aggregate throughput with very little or no unfairness among competing links. DCF, IDS and TES-E can only achieve one or the other and seldom both in the presence of errors.

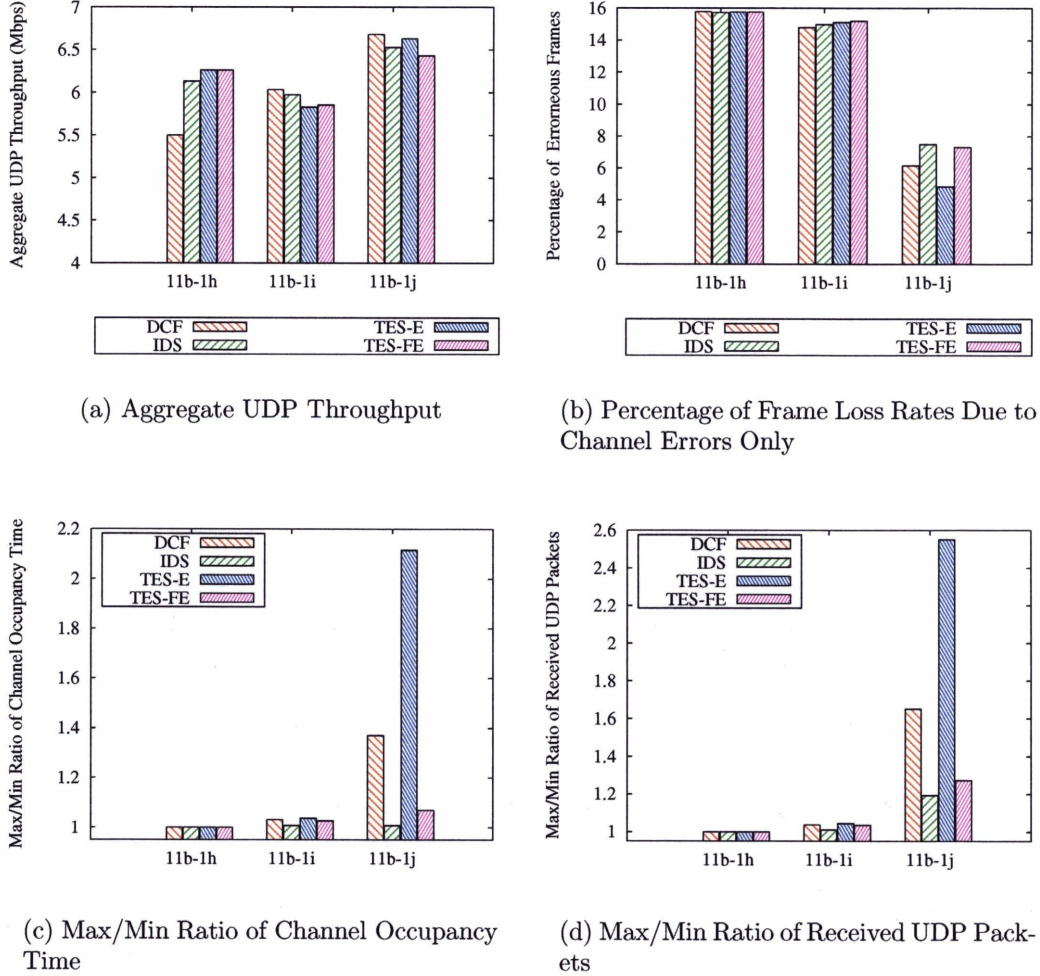


Figure 5-18: Measures related to efficiency and fairness when nodes experience random losses because of channel errors, under three different scenarios.

Figure 5-18 plots various measures under three different scenarios: 11b-1h, 11b-1i and 11b-1j. In scenario 11b-1h, there is only a single node transmitting to the AP while experiencing a bit error rate of 0.000014 (a roughly 16% frame loss rate, a high but realistic loss rate in practice.) In scenario 11b-1i, there are four nodes sending UDP data to a common AP, with each node experiencing a bit error rate of 0.000014. In scenario, 11b-1j, there are four nodes sending UDP data to a common AP; two nodes experience bit error rates of 0.000014 and the other two do not observe channel errors.

As shown in Figure 5-18(a), in 11b-1h, the achieved throughputs of the single node under TES-E and TES-FE are about 15% higher than that under DCF and the

achieved throughput under IDS is about 11% higher than under DCF. This is because a node under DCF increases its contention window size with each loss encountered (even when it is the only node occupying the channel). IDS outperforms DCF since nodes under IDS do not react to frame losses. TES-E and TES-FE outperform DCF since nodes under both schemes only react to losses when there is at least one other node competing for channel access. Note that the reduction in aggregate throughput is not limited to a single node case. Under a MAC protocol such as DCF that has a backoff scheme that simply reacts to losses by increasing average backoff delay, the aggregate throughput will be reduced whenever one or more nodes experience relatively high error rates but no other competing nodes have enough offered load to take advantage of additional channel occupancy time left unused by the nodes experiencing high loss rates.

Scenario 11b-1i examines the situation when all nodes experience similar error rates. In this scenario, the aggregate throughputs achieved under all four schemes are comparable (see 11b-1i in Figure 5-18(a)). Similarly, the max/min ratios of under all four schemes are also comparable, as shown in Figures 5-18(c) and 5-18(d).

In 11b-1j (half of the nodes experience high error rates), the aggregate throughputs achieved under both DCF and TES-E are slightly higher (3%) than those achieved under TES-FE and IDS. This is because under both DCF and TES-E, nodes with lower loss rates will achieve higher amount of channel occupancy time. Therefore, the overall frame loss rates attributable only to channel errors improve slightly under DCF and IDS as shown in Figure 5-18(b). However, Figures 5-18(c) and 5-18(d) show that the max/min ratios under DCF and TES-E are much higher than those under IDS and TES-FE. It is easy to see why IDS achieves better max/min ratios since it does not react to losses. However, TES-FE achieves similarly good max/min ratios even though nodes under TES-FE react to losses in a similar fashion to nodes under DCF and TES-E. This is because the fairness mechanism of TES-FE allows nodes that are lagging in channel occupancy time to achieve a higher share of channel occupancy time in the future. In other words, nodes that intentionally delay their transmissions by reacting to frame losses (to avoid potential burst losses) will soon “catch up” with other nodes when their observed channel conditions are better, leading to little unfairness.

In summary, in the presence of randomized channel errors:

- All schemes achieve similar aggregate throughput, and
- DCF and TES-E lead to significantly higher max/min ratios of channel occupancy time and received UDP packets than IDS and TES-FE.

The aggregate throughput achieved and the max/min ratios under TES-FE are comparable to those under IDS in the presence of random channel errors, even though TES-FE has a burst avoid scheme that reacts to losses.

5.7.7 Burst Losses

In this section, we explore the impact of losses that occur in bursts. We use a Rayleigh fast-fading model [80, 83] to capture the short time-scale fading phenomenon

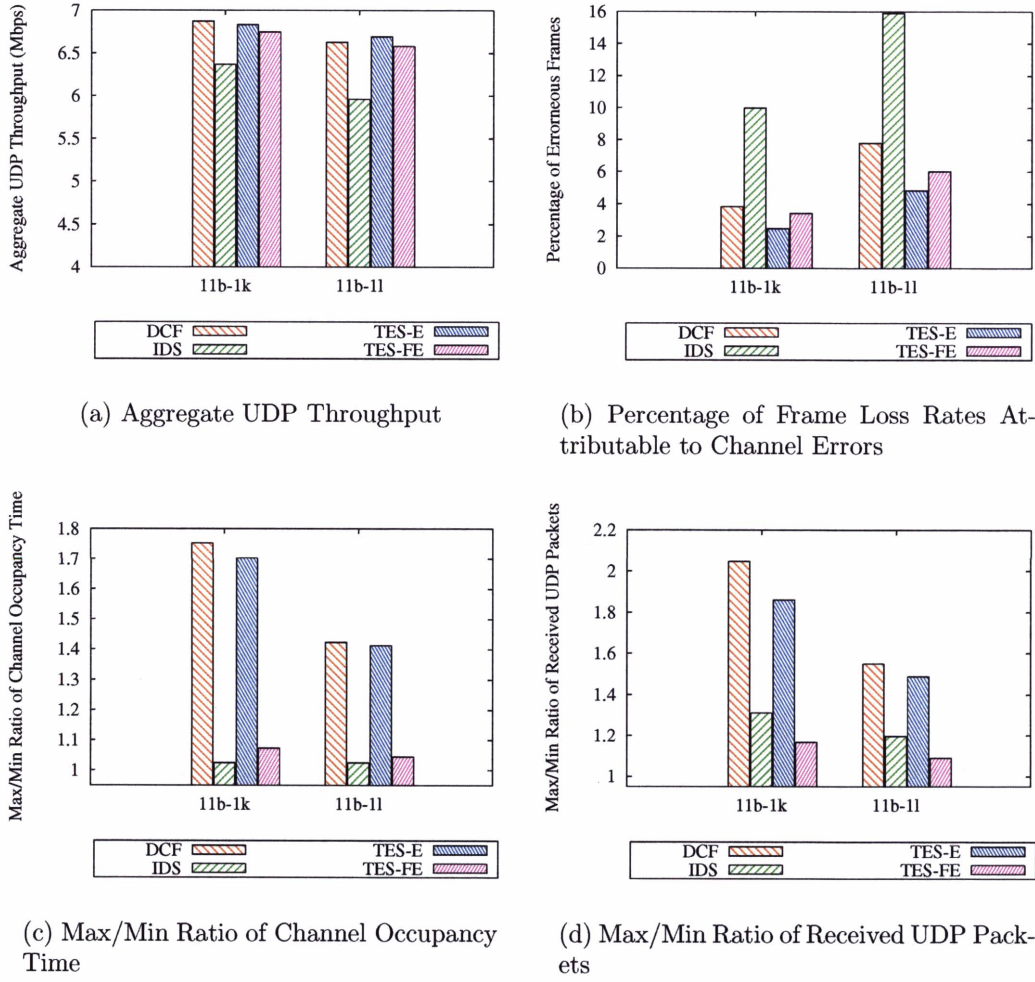


Figure 5-19: Measures related to efficiency and fairness when nodes experience losses in bursts because of channel errors, under two different scenarios.

that arises because of the movements of senders and/or receivers and those of other objects along transmission paths between transmitters and receivers. The received power thresholds for various data rates are based on the Orinoco 802.11b Gold Card data sheet [24].

We ran experiments under two scenarios: 11b-1k and 11b-1l. In both scenarios, four nodes send UDP data packets to a common AP at 11 Mbps. In scenario 11b-1k, two nodes are very close to the AP and two other nodes are about 80 and 90 feet away from the AP respectively. In scenario 11b-1l, the four nodes are at 60, 70, 80 and 90 feet away from the AP respectively.

As shown in Figure 5-19(a), in both scenarios, the aggregate throughputs achieved under IDS is less than those achieved under all other schemes. TES-FE achieves 6% and 10% higher in aggregate throughput than IDS under scenarios 11b-1k and 11b-1l respectively. This is because TES-FE reduces the frame loss rate attributable to

channel errors. As shown in Figure 5-19(b), TES-FE achieves 58% and 56% reductions aggregate frame loss rates (due to burst channel errors) over IDS. Specifically, in scenario 11b-11, the reduction of 10 percentage points in the aggregate frame loss rate (from 16% under IDS to about 6% under TES-FE) leads to a similar improvement in aggregate throughput. Such significant reductions in frame loss rates can lead to relatively more significant improvements in aggregate throughput in many scenarios when rate adaption mechanisms are used. Also, the observed collision rates under all schemes are similar (around 4.5%) for both scenarios (not shown in the figures).

Although DCF and TES-E achieve similar improvements in aggregate throughput and frame loss rates over IDS, they lead to significantly higher max/min ratios of channel occupancy time and received UDP packets than IDS and TES-FE.

To summarize:

- DCF, TES-E and TES-FE, by virtues of their simple mechanisms to avoid burst losses, improve aggregate throughput and significantly reduce overall loss rates in the presence of time correlated errors, when compared to IDS, which has no such mechanism,
- DCF and TES-E, however, lead to significantly higher max/min ratios of channel occupancy time and received UDP packets than either IDS or TES-FE, and
- TES-FE is the only scheme that achieves high aggregate throughputs while providing little or no unfairness in situations where losses occur in bursts.

The results in this section and the previous section show that the fairness mechanism of TES, in the presence of any types of errors, can provide a high degree of fairness even when burst avoidance schemes are used.

5.8 Summary

In this chapter, we described TES, a distributed MAC protocol, that achieves time-based, per-link fairness and scales well with the number of contenders. Unlike most existing protocols, TES does not rely on a spontaneous loss feedback mechanism, such as a stop-and-wait ARQ mechanism, to adjust CW . Instead, TES relies on idle time to adjust CW that is appropriate for the current contention level.

The default configuration of TES, TES-FE, includes a fairness mechanism and a simple burst avoidance scheme. TES-FE ensures that each competing entity achieves its fair share of channel occupancy time in the long term even in the presence of burst loss avoidance schemes, which voluntarily give up transmission opportunities to avoid burst losses.

Through extensive simulations, we showed that compared to DCF and IDS, TES-FE achieves high efficiency with little or no unfairness in the presence of rate diversity, random channel errors, time-correlated channel errors and many contenders.

Chapter 6

Centralized Time-based Regulator

The previous chapter describes and evaluates TES, a distributed MAC protocol that achieves time-based, per-link fairness and scalability. However, TES is not backward-compatible in a sense that it cannot be used in existing 802.11 networks that use DCF. In this chapter, we describe a backward-compatible link-layer solution that can provide the benefits of time-based fairness without requiring modifications at the MAC layer. Specifically, we:

- Present an effective backward-compatible scheme, TBR (for Time-based Regulator), for deploying time-based, per-duplink (or per-client) fairness in existing AP-based WLANs, irrespective of the MAC protocol used,
- Describe an efficient 802.11-based implementation of TBR that requires changing only the driver on the access point, and
- Demonstrate that TBR achieves time-based fairness by evaluating it on a Linux-based 802.11b testbed.

We also discuss the limitations of TBR. In particular, our TBR implementation does not alleviate many deficiencies of the underlying MAC protocol. For example, even with TBR, the collision rate of an 802.11b network will still be high when the number of contenders is more than a dozen. Furthermore, unlike TES, TBR requires coordination among multiple cells which may not be possible when they fall under different administrative domains. TBR is suitable when efficient coordination among multiple cells is possible and a backward-compatible solution is necessary. Most of the work in this chapter appear in [94].

6.1 TBR: Design and Implementation

TBR runs at each AP and provides an equal share of long-term channel occupancy time to each competing client node by

- Dictating how frame transmissions are scheduled at the AP as well as at the clients; TBR allows data exchange between the AP and a client only if the

channel occupancy time allocated in the past few seconds for exchanging data is not more than the client's fair,

- Taking into account the channel occupancy time of traffic in both downlink and uplink directions, and
- Taking into account varying traffic conditions, loss rates, data rates, and frame sizes.

TBR is based on a token bucket scheme [8]. Token bucket or leaky bucket based approaches are used to shape traffic in wired networks especially in ATM networks [18, 84, 104]. We develop a practical, adaptive token bucket based approach to provision channel occupancy time among competing clients.

The fundamental unit or token used in the implementation is the channel occupancy time in micro-seconds. TBR schedules the transmission of a frame destined to or originated from a client only if the node has not used up all its available channel occupancy time.

Figure 6-1 shows the pseudo-code of TBR that runs on the AP. TBR is part of a link-control sublayer and thus sits between the MAC sublayer and the network layer. TBR is implemented in five event handlers, each of which is triggered by the network layer, a timer or the MAC layer.

When a node i associates with the AP (i.e., joins the network), ASSOCIATEEVENT is triggered. The procedure i) creates output queue $queue_i$ and ii) initializes $tokens_i$, the available $tokens$, to an initial value, T_{init} . $T_{init} \leq Bucket$. $Bucket$ is a system-wide pre-configured parameter that specifies the maximum number of tokens that the node can accumulate. $rate_i$, the rate at which $tokens_i$ is being re-filled. The idea is that TBR will allow data exchange between the AP and client i only if $tokens_i > 0$, i.e., the client has not used up all of its fair share of channel occupancy time. Since $tokens_i$ is filled at $rate_i$, the amount of channel occupancy time allocated to client i for communicating with the AP will be capped at $rate_i$ in the long-term.

Whenever the upper layer has a packet p to transmit, it calls APPTXEVENT. TBR simply enqueues the packet to $queue_i$ where i is the destination of p .

TBR adjusts $tokens_i$ according to the channel occupancy time of transmitted frames originated from or destined to node i . Section 6.1.2 described how TBR computes the channel occupancy time. $Bucket$ determines the maximum length of the burst period in which node i can transmit successively (if no other nodes can transmit). $Bucket$ can affect the short-term fairness, an issue we discuss in Section 6.2.1.

TBR sets up a timer that periodically calls FILLEVENT. For each client i , the procedure updates $tokens_i$ using $rate_i$ and the total amount of available channel occupancy time t , since the last time FILLEVENT was called. Typically, t is less than $elapsed$, the elapsed time, because the channel is not always used for data transmissions; transmission events are typically separated by idle time, which should not be considered as part of the channel occupancy time.

$rate_i$ is the rate at which tokens are being re-filled. The summation $\sum_{i=1}^n rate_i = 1$, where n is the number of active client nodes. In general, $rate_i$ can vary among

client nodes depending on the desired fairness policy. If each competing node should receive an equal share of the channel occupancy time, $rate_i = \frac{1}{n}$. In practice, not all nodes can consume their available channel occupancy time according to the allocation. TBR ensures that the system remains work conserving by adjusting the token rates appropriately, as discussed in Section 6.1.3.

```

PROCEDURE ASSOCIATEEVENT(i) {
     $tokens_i \leftarrow T_{init}$ 
     $rate_i \leftarrow$  fair share of channel occupancy time
    initialize  $queue_i$ 
}
PROCEDURE APPTXEVENT(p) {
    i  $\leftarrow$  destination of p
    enqueue p to  $queue_i$ 
}
PROCEDURE FILLEVENT() {
     $elapsed \leftarrow$  the time elapsed since the last time FILLEVENT was called
     $t \leftarrow$  total amount of channel occupancy time during  $elapsed$ 
    for each link i
         $tokens_i \leftarrow \min(tokens_i + t * rate_i, Bucket)$ 
     $lastclock \leftarrow clock$ 
}
PROCEDURE MACTXEVENT() {
    for each link i starting with  $nexti$ 
        if  $queue_i$  is not empty and  $tokens_i > 0$ 
            dequeue a packet p from  $queue_i$ 
            ask the MAC to transmit p
             $nexti \leftarrow$  next link after i
}

```

Figure 6-1: Pseudo-code of most of TBR.

6.1.1 Scheduling Frame Transmissions

Whenever the MAC layer is ready to accept a new packet for transmission, it calls MACTXEVENT. TBR chooses one output queue among all the output queues with positive available channel occupancy time (tokens) and dequeues a packet for transmission.

The manner in which the output queue is chosen has no impact on the overall effectiveness of achieving fair time allocation in the long-term, since only the queues with positive tokens are considered. Nonetheless, the order could impact the short-term fairness. For simplicity and to alleviate short-term unfairness, TBR chooses the output queue among those with positive tokens in a round-robin manner. Short-term

unfairness can further be reduced by choosing the queue whose head-of-line packet has the shortest expected final completion time.

Once the output queue is chosen, TBR can decide which frame in the queue gets transmitted. For TCP, in-order packet delivery is desirable and thus a first-in-first-out discipline is preferable. However, if there are time-sensitive packets (used by real-time protocols), they should have priority over TCP packets with earlier arrival times. The correctness of TBR does not depend on how a packet to dequeue is chosen. We also note that TBR works with various buffering schemes (e.g. RED, drop-tail) that dictate which packets to drop when the queue is getting full. We distinguish buffering schemes from packet scheduling schemes. The former is responsible for deciding which packets to drop whereas the latter decides which packet gets transmitted [25].

TBR also dictates the scheduling of packet transmissions at the clients. Specifically, whenever $tokens_i \leq 0$, TBR needs to explicitly inform node i to delay transmission for a short amount of time. This can be accomplished in two ways. First, the TBR agent at the AP can inform the client by sending an explicit notification packet or piggybacking such information in a downlink packet. Second, the client can monitor the total channel occupancy time of packets transmitted and received, and transmit only if there is available channel occupancy time allocated for the node. To do so, the client needs to know only $rate_i$. However, as we explain in Section 6.1.3, TBR at the AP may update $rate_i$ depending on the overall traffic conditions and when that happens, TBR needs to inform the client. In both cases, a client agent is necessary at each client to communicate with TBR at the AP. We use the first method.

The amount of communication overhead depends on the MAC protocol used. TBR requires a single bit in the MAC header of a data frame transmission to notify the client to delay its transmission for a pre-determined amount of time. Even in cases where there is only uplink traffic, TBR can inform the client to delay its transmission with little overhead if the underlying MAC protocol (e.g. DCF) employs a *stop-and-wait* retransmission strategy. A stop-and-wait protocol requires the node receiving a data frame to reply with a synchronous acknowledgment, which can carry the TBR notification bit. Furthermore, if the underlying MAC protocol employs a polling mechanism (such as 802.11's PCF), no explicit communication is necessary since TBR can dictate which node gets polled.

Avoiding the Need for Client Agents

Although we just described how TBR cooperates with clients through client agents, cooperation from each client is only necessary if the client has uplink UDP flows that represent a significant fraction of its traffic. That is, client agents are unnecessary when TCP dominates WLAN traffic, which is typically the case as indicated by studies at university campuses [60, 96] and at a multi-day conference [7]. The studies show that TCP accounted for more than 90% of bytes exchanged over the WLANs.

Irrespective of the direction (uplink or downlink) of a TCP data packet stream of a client, regulating only frame transmissions from the AP to the client is sufficient to ensure that the client does not achieve more than its fair share of channel occupancy

time. TCP data packets are paced by TCP *ack* packets (“ack clocking” [45]) sent out by the receiver. In a typical scenario, all TCP data and *ack* packets go through the same AP. Therefore, delaying TCP data (*ack*) packets at the AP (by TBR) has the effect of slowing down the sending rates of downlink (uplink) TCP flows.

Thus, when TCP dominates WLAN traffic, an implementation of TBR that requires no modification to the underlying MAC protocol and to the drivers of mobile clients, is sufficient, thereby allowing incremental deployment and preserving backward compatibility. Through such a Linux-based implementation, we demonstrate in Section 6.2 that TBR without client agents can effectively provide long-term channel time guarantees for TCP flows in both directions as well as downlink UDP flows.

```

PROCEDURE COMPLETEEVENT( $p$ ) {
   $t \leftarrow$  channel occupancy time of  $p$ 
  if  $p$  was sent by AP
     $i \leftarrow$  destination of  $p$ 
  else
     $i \leftarrow$  source of  $p$ 
   $tokens_i \leftarrow tokens_i - t$ 
  if ( $actual_i = 0$ )
     $start_i \leftarrow$  current time
   $actual_i \leftarrow actual_i + t$ 
}

```

Figure 6-2: Pseudo-code of TBR that keeps track of the channel occupancy time allocated to each client.

6.1.2 Computing Channel Occupancy Time

Whenever the MAC layer has either finished sending or receiving packet p , it triggers COMPLETEEVENT. This procedure subtracts the channel occupancy time of p from the tokens associated with the node, i , that is the source or destination of p . It also modifies $actual_i$, the actual tokens used since $start_i$. We will explain how TBR uses $actual_i$ in the next subsection.

We now describe how to compute the channel occupancy time for packet p . To send a packet, multiple frame transmissions (retransmissions) at the MAC-layer may be necessary because of losses. Failed frame transmissions also contribute to the channel occupancy time required to send a packet. Therefore the channel occupancy time required to send or receive a packet is the sum of the channel occupancy time required at the MAC layer for each frame transmission until p has successfully been transmitted or dropped as a result of an undeliverable failure.

Taking into account retransmissions is straight forward in the downlink direction. However, in the uplink direction, the AP is not aware of the exact number of retransmission attempts made by the client stations. Ideally, the underlying MAC protocol

should include a retry sequence number field in the header to indicate how many re-transmissions precede the current packet transmission. However, the existing version of the DCF's MAC header only includes a 1-bit retry field to indicate whether the frame is a retransmission or not.

When retransmission information is not available for each packet received and the necessary header modification is not an option, the AP needs to estimate the information necessary to approximate the channel occupancy time. We distinguish two types of losses at the AP: one detected at the MAC layer (by a CRC check failure) and the other at the physical layer. In the former, it is highly likely that the MAC header, whose size is relatively much smaller than the typical payload size, is not corrupted and thus the AP can determine the source address of the failed transmission as well as the transmission rate. The MAC layer header can be made robust against channel errors by transmitting at a lower data rate.

However, if the frame loss is detected at the physical layer, TBR can be aware of the loss but may not know the source of the transmission and the link-layer sequence number. We believe that heuristics can be developed to estimate the transmission information of each loss detected at the physical layer based on i) the number of active clients in the last few dozen milliseconds and ii) their steady state loss rates at the downlink direction. We have not developed such a mechanism and we discuss the consequences in Section 6.1.4.

6.1.3 Keeping Channel Utilization High

When traffic contains a mixture of TCP and UDP flows that have various sending rates (and bottleneck link bandwidth), it is important to correctly determine the amount of channel occupancy time made available to each node. Specifically, TBR needs to adjust $rate_i$ to reflect changing traffic conditions. For instance, the system will be under-utilized if we give each node $\frac{1}{n}$ of the available channel occupancy time but some nodes cannot consume all of their available time shares whereas others could consume more if allowed.

TBR periodically adjusts $rate_i$ associated with each node i so that the channel utilization is kept high without violating the max-min fairness constraint [19, 46].

DCF in conjunction with a simple round-robin queuing scheme at the AP generally achieves the max-min notion of fairness when only TCP flows are involved. Assume that there are 3 uplink TCP flows and that one flow can only consume $\frac{1}{5}$ of the channel bandwidth (the wireless hop is not its bottleneck link). DCF will allow each of the remaining flows to consume $\frac{2}{5}$ of channel bandwidth provided that the bottleneck link of both flows is the wireless link.

TBR with any MAC protocol achieves the same fairness criteria provided that each client node with data to transmit contends for channel access whenever it has data to transmit. Satisfying the max-min fairness criteria does not require that the actual demand of each node is known. Rather, one can achieve the fairness goal by incrementally giving more channel occupancy time to each competing node that can consume all the channel occupancy time made available to it [8] and ensuring that no nodes receive more channel occupancy time than they can consume. We implement

this general idea in TBR.

```

PROCEDURE ADJUSTRATEEVENT() {
  for each node  $i$ 
     $excess \leftarrow rate_i - \frac{actual_i}{now-start_i}$ 
    if ( $excess \leq R^{thresh}$ )
      if  $excess < E^{min}$ 
         $E^{min} \leftarrow excess$ 
      if  $excess > E^{max}$ 
         $m \leftarrow i$ 
    else
      add  $i$  to set  $I'$ 
   $E^{min} \leftarrow \frac{E^{min}}{2}$ 
  for each node  $j \in I'$ 
     $rate_j \leftarrow rate_j + \frac{E^{min}}{|I'|}$ 
   $rate_m \leftarrow rate_m - E^{min}$ 
  for each node  $j \in I$ 
     $actual_j \leftarrow 0$ 
}

```

Figure 6-3: Pseudo-code of the token rate adjustment event

Initially, each competing node starts with the desired token rate of $\frac{1}{n}$. TBR schedules a timer event called ADJUSTRATEEVENT that periodically adjusts the token *rate* available to each node. As shown in Figure 6-3, ADJUSTRATEEVENT computes the excess capacity of the *under-utilized nodes*, i.e., nodes for which the actual token rate ($actual_i$) is lower than the assigned rate by the threshold R^{thresh} . It then computes the excess capacity E^{min} to redistribute equally among nodes (I') that have fully utilized the provisioned bandwidth in the previous round.

The actual method of computing E^{min} is of little importance for the long-term correctness so long as E^{min} is not too big. However, E^{min} does affect the responsiveness of TBR to changing traffic conditions in the short-term. Briefly, if E^{min} is too large, the instantaneous throughputs experienced by flows can significantly vary. Such behaviors may increase the buffer requirements at the nodes to avoid TCP *ack* compression that can lead to packet drops. If E^{min} is too small, it can take a long time until a max-min channel occupancy time allocation is reached. Meanwhile, the channel may be under-utilized since some nodes that have data to send are not allowed to transmit even though there is excess channel occupancy time left unused by some other nodes.

Figure 6-3 shows one way of choosing E^{min} . We pick, among all under-utilized nodes, node m with the maximal excess capacity (the largest difference in actual and assigned token rate). Half of E^{min} is subtracted from m 's token rate and redistributed among nodes that have consumed tokens at rates close to their assigned rates. In

Section 6.2, we show that using this mechanism TBR is able to keep the channel utilization high in the presence of varying traffic conditions.

6.1.4 An 802.11-based Implementation

We implemented TBR in the HostAP [50] driver running on a Linux PC as a proof of concept. The HostAP driver implements access point functionality so that PCs equipped with a Prism chipset based 802.11 cards can act as APs. We use unique 6-byte MAC addresses as node identifiers.

TBR requires APs to set up per-node output queues. However, the total buffer space requirement is comparable between a normal AP and an AP with TBR. For instance, if an existing AP has the total queue size of x packets than a TBR-equipped AP can setup n queues each with $\frac{x}{n}$ packets, where n is the number of competing nodes. For ease of implementation, our TBR implementation uses FIFO queues. As explained before, TBR can work with any buffering scheme.

We set up a timer that is invoked every 1 to 2 milliseconds (depending on how busy the system is handling hardware interrupts) to call `FILLEVENT` and `ADJUSTRATEEVENT`. Also, we set $T_{init} = 50$ ms and $Bucket = 100$ ms.

Finally, the current implementation of TBR does not use the retransmission information in computing the packet transfer time. Thus, TBR in some cases can cause slight biases in granting channel occupancy time to competing nodes. Nonetheless, as we show in Section 6.2, it does well in achieving its goal.

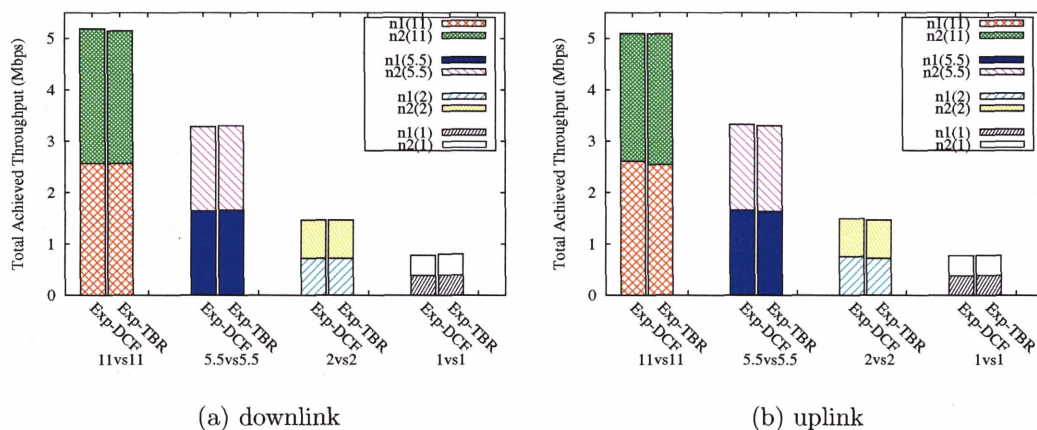


Figure 6-4: TCP throughputs achieved in either uplink or downlink direction by two competing nodes using the same data rate. *Exp-DCF* and *Exp-TBR* denote the experiments that were run with the AP equipped without or with TBR respectively. $n_i(11)$ denotes the throughput achieved by node i transmitting at 11 Mbps.

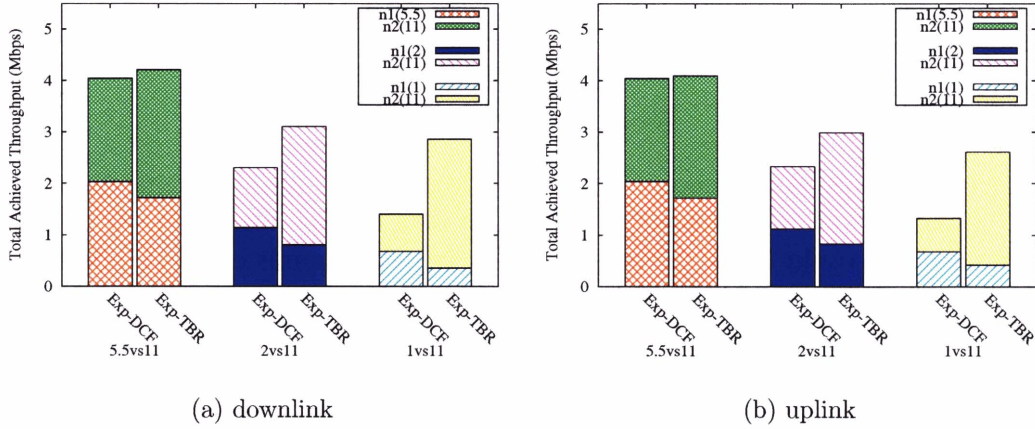


Figure 6-5: TCP throughputs achieved in either uplink or downlink direction by two competing nodes using different data rates. *Exp-DCF* and *Exp-TBR* denote the experiments that were run with the AP equipped only with DCF without TBR and DCF with TBR respectively.

6.2 Evaluation

We setup experiments to evaluate the correctness and performance of TBR. We used a PIII-700MHz Linux laptop equipped with a D-Link DWL-650 card running the HostAP driver as the AP and IPAQs equipped with Cisco-350 cards as competing nodes.

For each type of experiment, we ran two different AP configurations: one with TBR, *Exp-TBR*, and one without, *Exp-DCF*. Each data point is an average of 5 runs and in each run, each contending node sends about 2000 1500-byte packets. All throughputs measured are achieved TCP throughputs.

When the AP is run under the *Exp-DCF* configuration, no queue is set up in the driver. Instead, the kernel interface queue (with the maximum size of 110) is used to store packets. When the AP is run with TBR, n queues each with the maximum queue size of $\frac{100}{n}$ are set up inside the driver. The kernel interface queue is then set to 10. Thus, the total buffer space available to each scheme is the same.

Figure 6-4 compares the throughputs achieved by two competing nodes using the same data rate when the AP is configured with or without TBR. When competing nodes use the same data rate, *Exp-TBR* and *Exp-DCF* yield almost identical results, showing that TBR incurs little overhead.

When nodes use different data rates, the throughput achieved by each competing node as well as the total throughput differ significantly depending upon whether TBR is used or not. As shown in Figure 6-5(a), when TBR is used, the total achieved throughput in the downlink direction increases by about 6% in the 5.5vs11 case, 35% in the 2vs11 case and 103% in the 1vs11 case.

Figure 6-5(b) shows similar improvements achieved by TBR in the uplink direction.

Throughput	Exp-DCF	Exp-TBR
n1	2.9434	2.9542
n2	2.1276	2.1193
Total	5.071	5.061

Table 6.1: Comparison of achieved TCP throughputs under Exp-DCF and Exp-TBR. Node $n2$ experienced the bottleneck bandwidth of 2.1 Mbps whereas node $n1$ could send as fast as it could (TCP permitted). Both nodes transmitted at 11 Mbps.

To understand how TBR works when traffic contains flows with various demands, we set up a scenario that involved two nodes, $n1$ and $n2$, each sending TCP packets at the same data rate of 11 Mbps but experienced different bottleneck link capacities. $n2$ experienced the bottleneck bandwidth of 2.1 Mbps while the wireless link is $n1$'s bottleneck. We achieved this by limiting the sending rate of the application generating TCP packets at $n2$. The expected DCF behavior is to give $n2$ 2.1 Mbps of channel bandwidth and $n1$ the remainder. Table 6.1 shows the throughputs achieved under *Exp-TBR* and *Exp-DCF*. There is no significant difference between the two sets of results, showing that the rate adjustment algorithm described in Section 6.1.3 works.

6.2.1 Limitations and Extensions

TBR is currently intended for ensuring that each competing node receives an equal share of channel occupancy time based on max-min fairness over the long run. As we later demonstrate in Section 6.2, TBR works well when competing flows last for hundreds of packets. We believe that long-lived flows (e.g., file transfer applications) are usually the cause of congestion in enterprise and university networks, congestion in *hot-spot*.

However, TBR may not provide fair channel occupancy time allocations among clients in the short term, thus mostly affecting flows that are relatively short-lived (in the order of tens of packet transfer time). This is because our $rate_i$ adjustment process may not be responsive enough to reflect changing demands in the short-term. Responsiveness of TBR relies on i) the manner with which it adjusts the token rate assigned to each competing node and ii) the frequency of adjustment, i.e., how often ADJUSTRATEEVENT) is called. We did not explore TBR's responsiveness on short time-scales.

TBR can be modified to provide each competing node with the desired share of channel occupancy time (not necessarily equal). Therefore, QoS mechanisms may use TBR to provide QoS at existing AP-based WLANs. Furthermore, although the current implementation of TBR allocates channel occupancy time to nodes, it can be extended to allocate channel occupancy time among flows.

The 802.11e standard [44] currently being drafted defines quality of service support for the 802.11 MAC. Using 802.11e, competing nodes acquire time-limited transmission opportunities, each of which is defined as an interval of time when a station has

the right to initiate transmissions. Time-limited TXOPs are allocated via contention or granted through a centralized coordinator. Furthermore, 802.11e differentiates the probability of channel access based on the traffic categories. TBR can be integrated with 802.11e by choosing appropriate traffic categories for each competing node according to their fair share of channel occupancy time.

6.3 Summary

In this Chapter, we described a practical link-layer scheduling scheme called TBR that works in conjunction with 802.11's DCF to provide long-term time-based fairness in AP-based WLANs. TBR uses a token bucket based approach to provision channel occupancy time among nodes communicating via an AP. We showed that TBR can be implemented in an AP driver in a way that is backwards compatible with existing 802.11 standard. We implemented our scheme in the Linux HostAP driver running on a PC used as the AP, and evaluated it through a series of experiments. In the absence of rate diversity, the performance of our implementation is equivalent to the standard implementation. In the presence of rate diversity, it achieves the predicted gains.

Chapter 7

Conclusions

In this chapter, we summarize the work presented in this dissertation, discuss future directions and end with concluding thoughts.

7.1 Summary

We began by showing that today's distributed MAC protocols lead to low aggregate throughput and high delay, and noticeable unfairness in the presence of rate diversity, varying channel conditions, non-cooperative rational competition, and many contenders.

Through analyses, experiments and simulations, we identified the root causes of poor efficiency and unfairness. Specifically, we show that frame-based or bit-based fairness used by many existing MAC protocols, including 802.11's DCF, leads to poor efficiency and the use-it-or-lose-it policy of DCF leads to a high degree of unfairness. We also show that protocols like DCF do not scale with the number of contenders. Furthermore, we show through a game theoretic analysis and simulation that a combination of frame-based fairness and the use-it-or-lose-it policy can often lead rational, non-cooperative nodes to employ inefficient transmission strategies at equilibria.

- We then argued that the fundamental shared resource for a given wireless channel is *channel occupancy time*, the time available to transfer data, and not bits or frame transmission opportunities.
- This led us to argue for time-based fairness, under which competing entities with equal priorities achieve equal shares of channel occupancy time. Through analysis, we showed that time-based fairness improves the overall network performance over many traditionally accepted fairness notions, including frame-based fairness.
- We also argued that a MAC protocol should provide long-term fair channel occupancy time shares guarantees among competing entities. Through a game theoretic analysis and simulation, we showed that a MAC protocol that provides

long-term channel occupancy time shares guarantees can lead rational nodes to more efficient equilibria than a MAC protocol, such as DCF, that only allocates channel occupancy time on a use-it-or-lose it basis.

We developed two solutions, TES and TBR, that achieve time-based fairness and provide fair channel occupancy time shares guarantees for two different target environments.

TES Our Time-fair Efficient and Scalable MAC protocol is a distributed non backward-compatible MAC protocol that achieves high network efficiency with little unfairness in both long and short timescales—attributes that have not been achieved simultaneously by existing distributed MAC protocols. Our simulation results show that compared to DCF, TES i) in the presence of rate diversity, improves by as much as 170% and 140% in aggregate UDP and TCP throughput respectively, ii) improves aggregate UDP throughput by as much as 70% when the number of contenders is in several dozens, ii) reduces frame loss rates by as much as 60% in the presence of time-correlated channel errors, and iii) achieves a noticeably higher or no-less degree of fairness in all cases that we tried.

TBR Our Time-based Regulator is a backward-compatible link-layer scheduler that runs at the AP, works in conjunction with DCF, and requires no modifications to clients nor to DCF. TBR is appropriate for existing AP-based networks but, unlike TES, is ineffective when nearby non-cooperative nodes fall under different administrative domains. An evaluation of our Linux implementation of TBR shows that TBR is effective in providing fair channel occupancy time allocation among clients and can improve aggregate TCP throughput by as much as 105% in the presence of rate diversity.

7.2 Potential Future Research Work

We now discuss future directions of the work presented in this dissertation.

1. *More Efficient ARQ*: TES decouples its collision control mechanism from the link-layer retransmission mechanism. Since link-layer feedback (of loss) is no longer necessary for adjusting *CW* to achieve a value that is appropriate with the contention level, it is no longer essential to use a stop-and-wait ARQ protocol like the one used by DCF. TES can work well with a more efficient retransmission mechanism like selective-repeat that requests retransmissions only of frames that are not correctly received. However, a responsive feedback mechanism may still be necessary for burst avoidance schemes. Therefore, the challenge is to develop a retransmission mechanism that takes advantage of TES's collision control mechanism, which does not require feedback, while responsively informing the sender to delay its transmissions in the presence of time-correlated errors.
2. *More Effective Rate Adaptation*: TES achieves a bounded collision rate. Therefore, each node can deduce its frame loss rate attributable to channel errors

from the observed loss rate and the known collision rate. By being able to accurately determine its error rate, a node may be able to select an appropriate data transmission rate more effectively.

3. *Time-based, Power-based Fairness*: Our game theoretic analysis does not consider choosing transmit power as part of the transmission strategy space. However, transmit power is an important criteria, since the higher the transmit power of a frame transmission, the higher the probability of that frame being received successfully, and the more noise level it adds to nearby transmissions. It will be a natural evolution to develop a fairness notion in terms of both channel occupancy time and transmit power used.
4. *Spectrum Sharing Etiquette*: Our game theoretic analysis showed that a MAC-layer fairness notion is critical in achieving efficiency in the presence of rational non-cooperative competition. Since many heterogeneous WLAN technologies share unlicensed frequency bands, it is important to establish a set of etiquette for fair and efficient sharing of spectrum. Although we showed that a CSMA MAC protocol can achieve time-based fairness in a distributed manner, it is not clear how best to achieve a similar goal when competing devices run different MAC protocols, including non-contention based protocols.

7.3 Conclusion

Our work examines the impact of MAC-layer fairness notions on network efficiency. We presented our work in the context of 802.11 WLANs. However, most of our work is relevant for any contention-based distributed MAC protocol. We believe that most future WLAN technologies operating in unlicensed bands will at least in part contend for channel access in a distributed fashion without explicit coordination, even when all nodes fall under the same administrative domain. The reason is because it is almost impractical to coordinate channel access in a centralized manner among competing devices running a disparate set of WLAN technologies, including Bluetooth, designed for short-range low-power communications, cordless phones, and Zigbee, a newer standard to support sensor-based ultra-low-power communications, especially when data exchange among devices often occurs in bursts.

Our solution, TES, demonstrates that efficiency and scalability can be achieved in a distributed contention-based fashion with little or no sacrifice to fairness. We hope that our analyses, observations and the techniques used in TES will help future standard bodies, including the 802.11 standard bodies, develop next generation of MAC protocols that are suitable for multi-rate WLANs and non-cooperative heterogeneous environments. The wireless revolution is on. It is important that we at least maintain and possibly improve aggregate utilities of WLAN users, whose productivity will increasingly depend on robust wireless communications.

Bibliography

- [1] IEEE Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: High Speed Physical Layer in the 5 GHz Band, September 1999.
- [2] IEEE 802.11b/d3.0 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, August 1999.
- [3] N. Abramson. The Aloha system - another alternative for computer communications. In *Proceedings of Fall Joint Computer Conference, AFIPS Conference*, 1970.
- [4] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In *Proceedings of SIGCOMM*, Portland, Or, August 2004.
- [5] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani, and R. D. Gitlin. AIR-MAIL: A link-layer protocol for wireless networks. *ACM Wireless Networks*, 1(1):47–60, 1995.
- [6] Anand Balachandran, Geoffrey Voelker, and Paramvir Bahl. Hot-spot congestion relief in public-area wireless networks. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, Callicoon, NY, June 2002.
- [7] Anand Balachandran, Geoffrey M. Voelker, Paramvir Bahl, and P. Venkat Rangan. Characterizing user behavior and network performance in a public wireless LAN. In *Proceedings of SIGMETRICS*. ACM Press, June 2002.
- [8] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, second edition, 1992.
- [9] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LANs. In *Proceedings of SIGCOMM*, pages 212–225, September 1994.
- [10] G. Bianchi, L. Fratta, and M. Oliveri. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In *Proceedings of PIMRC*, October 1996.
- [11] The specification of the Bluetooth system. <http://www.bluetooth.com/>, December 1999. Bluetooth Special Interest Group document.

- [12] L. Bononi, M. Conti, and E. Gregori. Design and performance evaluation of an asymptotically optimal backoff algorithm for IEEE 802.11 wireless LANs. In *Proceedings of HICSS*, page 8025, Washington, DC, USA, 2000. IEEE Computer Society.
- [13] Luciano Bononi, Marco Conti, and Enrico Gregori. Runtime optimization of IEEE 802.11 wireless LANs performance. *IEEE Transactions on Parallel and Distributed Systems*, 15(1):66–80, 2004.
- [14] J. Bruno, E. G. Coffman, and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM*, 17:382–387, Jul 1974.
- [15] F. Cali, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE Transactions on Networking*, 8(6):785–799, December 2000.
- [16] Zhiruo Cao, Zheng Wang, and Ellen W. Zegura. Rainbow fair queueing: Fair bandwidth sharing without per-flow state. In *Proceedings of INFOCOM*, pages 922–931, 2000.
- [17] John Ippocratis Capetanakis. *The Multiple Access Broadcast Channel : Protocol and Capacity Considerations*. PhD dissertation, Massachusetts Institute of Technology, Cambridge, Department of Electrical Engineering and Computer Science, December 1977.
- [18] H. J. Chao. Design of leaky bucket access control schemes in ATM networks. In *Proceedings of ICC*, June 1991.
- [19] Dah-Ming Chiu and Raj Jain. Analysis of the increase/decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17(1):1–14, June 1989.
- [20] Sunghyun Choi, Youngkyu Choi, and Inkyu Lee. IEEE 802.11 MAC-level FEC with retransmission combining. *IEEE Transactions on Wireless Communication*, January 2005.
- [21] Mooi Choo Chuah, Bharat T. Doshi, Subrahmanyam Dravida, Richard Ejzak, and Sanjiv Nanda. Link layer retransmission schemes for circuit-mode Sata over the CDMA physical channel. *Mobile Networks and Applications*, 2(2):195–211, 1997.
- [22] Data Sheet of Cisco Aironet 350 Series Access Points. http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/carto_in.ht%a.
- [23] Data Sheet of Proxim ORiNOCO AP-200 Access Point. <http://www.proxim.com/learn/library/datasheets/AP-200.pdf>.
- [24] Data Sheet for Proxim ORiNOCO 11b/g PC Card. www.proxim.com/learn/library/datasheets/11bgpccard.pdf.

- [25] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Internetworking: Research And Experience*, 1:3–26, April 1990.
- [26] NetGear: Super G and Super A/G Modes. http://support.dlink.com/faq/view.asp?prod_id=1423, 2005.
- [27] High capacity multi-channel intelligent wideband WLAN technology. http://www.engim.com/products_en3001.html, 2005.
- [28] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of MOBICOM*, pages 263–270, August 1999.
- [29] Z. Fang, B. Bensaou, and Y. Wang. Performance evaluation of a fair backoff algorithm for IEEE 802.11 DFWMAC. In *Proceedings of MOBICOM*, pages 48–57, 2002.
- [30] G. Fayolle and I. Mitrani. Sharing a processor among many job classes. *Journal of the Association for Computing Machinery*, 27(3):519–532, Jul 1980.
- [31] Chane L. Fullmer and J. J. Garcia-Luna-Aceves. Floor Acquisition Multiple Access (FAMA) for packet-radio networks. In *Proceedings of SIGCOMM*, pages 262–273, New York, NY, USA, 1995. ACM Press.
- [32] Chane L. Fullmer and J. J. Garcia-Luna-Aceves. Solutions to hidden terminal problems in wireless networks. In *Proceedings of SIGCOMM*, pages 39–49, New York, NY, USA, 1997. ACM Press.
- [33] Xia Gao, Thyagarajan Nandagopal, and Vaduvur Bharghavan. On improving the performance of utility-based wireless fair scheduling through a combination of adaptive FEC and ARQ. *Journal of High Speed Networks*, 10(1):19–36, 2001.
- [34] Rodrigo Garces and J. J. Garcia-Luna-Aceves. Floor acquisition multiple access with collision resolution. In *Proceedings of MOBICOM*, pages 187–197, New York, NY, USA, 1996. ACM Press.
- [35] Robert Gibbons. *Game Theory for Applied Economists*. Princeton University Press, 1992.
- [36] S. Jamaloddin Golestani. A self-clocked fair queueing scheme for broadband applications. In *Proceedings of INFOCOM*, pages 636–646, 1994.
- [37] P. Goyal, H. M. Vin, and H. Cheng. Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks. *IEEE/ACM Transactions on Networking*, oct 1997.
- [38] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient routing protocols for wireless microsensor networks. In *Proceedings of HICSS*, January 2000.

- [39] Martin Heusse, Franck Rousseau, Gilles Berger-Sabbatel, and Andrzej Duda. Performance anomaly of 802.11b. In *Proceedings of INFOCOM*, April 2003.
- [40] Martin Heusse, Franck Rousseau, Romaric Guillier, and Andrzej Duda. Idle Sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs. In *Proceedings of SIGCOMM*, August 2005.
- [41] High Performance Radio Local Area Network (HIPERLAN) Type 1: Functional Specification, 1996.
- [42] Gavin Holland, Nitin H. Vaidya, and Paramvir Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *Proceedings of MOBICOM*, pages 236–251, 2001.
- [43] HomeRF Project. <http://www.homrf.org>, 1999.
- [44] IEEE 802.11 Working Group. Draft Supplement to International Standard for Information Exchange between systems - LAN/MAN Specific Requirements, November 2001.
- [45] Van Jacobson. Congestion avoidance and control. In *Proceedings of SIGCOMM*, pages 314–329, Stanford, CA, August 1988.
- [46] Ranjendra Jain, Dah-Ming Chiu, and William Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer system. Technical Report 301, Digital Equipment Corporation, September 1984.
- [47] Ranjendra Jain, Dah-Ming Chiu, and William Hawe. Congestion avoidance in computer networks with a connectionless network layer: Concepts, goals and methodology. Technical Report 506, Digital Equipment Corporation, 1987.
- [48] K. Jamieson, H. Balakrishnan, and Y. Tay. Sift: A MAC protocol for event-driven wireless sensor networks. Technical Report 894, MIT LCS, May 2003.
- [49] Kyle Jamieson, Bret Hull, Allen K. Miu, and Hari Balakrishnan. Understanding the real-world performance of carrier sense. In *ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND)*, Philadelphia, PA, August 2005.
- [50] Jouni Malinen. Host AP driver for Intersil Prism2/2.5/3. <http://hostap.epitest.fi>, 2003. Version 0.0.1.
- [51] A. Kamerman and L. Monteban. WaveLAN II: a high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, pages 118–133, Summer 1997.
- [52] P. Karn. MACA: A new channel access method for packet radio. In *Proceedings of ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 1990.

- [53] P. Karn. The Qualcomm CDMA digital cellular system. In *Proceedings of the USENIX Mobile and Location-Independent Computing Symposium*, pages 35–39, 1993.
- [54] Jae Hyun Kim and Jong Kyu Lee. Performance of carrier sense multiple access with collision avoidance protocols in wireless LANs. *Wireless Personal Communications*, 11(2):161–183, 1999.
- [55] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: Part i — carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23:1400–1416, 1975.
- [56] Leonard Kleinrock. *Queueing Systems, Volume I. Queueing Systems*, volume 1. John Wiley and Sons, 1975.
- [57] Leonard Kleinrock. *Queueing Systems, Volume II. Computer Applications*, volume 2. John Wiley and Sons, 1976.
- [58] Leonard Kleinrock. Time-shared systems: a theoretical treatment. *Journal of the ACM*, 14(2):242–261, 1967.
- [59] Can E. Koksal, Hisham I. Kassab, and Hari Balakrishnan. An analysis of short-term fairness in wireless media access protocols. In *Proceedings of SIGMETRICS*, June 2000.
- [60] David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. In *Proceedings of MOBICOM*. ACM Press, September 2002.
- [61] David Kotz, Calvin Newport, and Chip Elliott. The mistaken axioms of wireless-network research. Technical Report TR2003-467, Dept. of Computer Science, Dartmouth College, July 2003.
- [62] Xin Liu, Edwin K. P. Chong, and Ness B. Shroff. Transmission scheduling for efficient wireless network utilization. In *Proceedings of INFOCOM*, pages 776–785, 2001.
- [63] Yonghe Liu and Edward Knightly. Opportunistic fair scheduling over multiple wireless channels. In *Proceedings of INFOCOM*, Mar-Apr 2003.
- [64] Songwu Lu, Vaduvur Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, 1999.
- [65] Stefan Mangold. *Analysis of IEEE 802.11e and Application of Game Models for Support of Quality-of-Service in Coexisting Wireless Networks*. PhD thesis, Aachen University, Jun 2003.
- [66] Stefan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertz, and Lothar Stibor. IEEE 802.11e wireless LAN for quality of service. In *European Wireless*, volume 1, pages 32–39, February 2002.

- [67] Paul McKenney. Stochastic fairness queuing. In *Proceedings of INFOCOM*, June 1990.
- [68] R. M. Metcalfe and D. R. Boggs. ETHERNET: Distributed packet switching for local computer networks. *Communications of the ACM*, pages 395–403, 1976.
- [69] Vlad Mitlin. Optimal MAC packet size in networks without cut-through routing. *IEEE Transactions on Wireless Communications*, 2(5):901–910, September 2003.
- [70] Vlad Mitlin. Optimal selection of ARQ parameters in QAM channels: Research articles. *Wireless Communicaitons and Mobile Computing*, 5(2):165–174, 2005.
- [71] Allen Miu, Godfrey Tan, and Hari Balakrishnan. Divert: fine-grained access point selection for wireless LANs to reduce packet losses. In *Proceedings of MOBISYS*, Boston, MA, Jun 2004.
- [72] Allen K. Miu, Hari Balakrishnan, and Can E. Koksal. Improving loss resilience with multi-radio diversity in wireless networks. In *Proceedings of MOBICOM*, Cologne, Germany, September 2005.
- [73] Eytan Modiano. An adaptive algorithm for optimizing the packet size used in wireless ARQ protocols. *Wireless Networks*, 5(4):279–286, 1999.
- [74] S. Nanda, R. Eljak, and B. T. Doshi. A retransmission scheme for circuit-mode data on wireless links. *IEEE Journal on Selected Areas in Communications*, 12(8):1338–1352, October 1994.
- [75] NetGear: Super G Mode. <http://www.netgear.com/applications/home/superg.php>, 2005.
- [76] ns-2 Network Simulator. <http://www.isi.edu/vint/nsnam/>, 2000.
- [77] Y. Ohtani, N. Kawahara, Y. N. Kawahara, H. Nakaoka, T. Tomaru K. Maruyama, T. Chiba, T. Onoye, and I. Shirakawa. Wireless digital video transmission system using IEEE 802.11b PHY with error correction block based ARQ protocol. *IEICE Transactions on Communications*, E85-B(10):2032–2043, October 2002.
- [78] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, Apr 1994.
- [79] Abhay K. Parekh and Robert G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single node case. In *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies on One world through Communications (Vol. 2)*, pages 915–924. IEEE Computer Society Press, 1992.

- [80] R. Punnoose, P. Nikitin, and D. Stancil. Efficient simulation for Ricean fading within a packet simulator. In *Proceedings of VTC*, pages 764–767, 2000.
- [81] X. Qiu and J. Chuang. Link adaptation in wireless data networks for throughput maximization under retransmissions. In *Proceedings of ICC*, 1999.
- [82] Parameswaran Ramanathan and Prathima Agrawal. Adapting packet fair queueing algorithms to wireless networks. In *Proceedings of MOBICOM*, pages 1–9, 1998.
- [83] T.S. Rappaport. *Wireless Communications*. Prentice Hall, Upper Saddle River, N.J., 1996.
- [84] Jennifer Rexford, Flavio Bonomi, Albert G. Greenberg, and Albert Wong. A scalable architecture for fair leaky-bucket shaping. In *Proceedings of INFOCOM*, pages 1054–1062, 1997.
- [85] L. Roberts. Aloha packet system with and without slots and capture. ASS Note 8, ARPA, Stanford Res. Insts, June 1972.
- [86] Raphael Rom and Moshe Sidi. Multiple access protocols: Performance and analysis (book review). *SIGMETRICS Performance Evaluation Review*, 20(3):5–6, 1993.
- [87] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic media access for multirate ad hoc networks. In *Proceedings of MOBICOM*, sept 2002.
- [88] Seattle Wifi map project. <http://depts.washington.edu/wifimap/>, 2005.
- [89] C.E. Shannon. A mathematical theory of communications. *Bell System Technical Journal*, 27:379–423 and 623–656, Jul and Oct 1948.
- [90] M. Shreedhar and George Varghese. Efficient fair queueing using deficit round robin. In *Proceedings of SIGCOMM*, August 1995.
- [91] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queueing: a scalable architecture to approximate fair bandwidth allocations in high speed networks. In *Proceedings of SIGCOMM*, 1998.
- [92] Godfrey Tan and John Guttag. Capacity allocation in wireless LANs. Technical Report 973, MIT CSAIL, Cambridge, MA, nov 2004.
- [93] Godfrey Tan and John Guttag. Long-term time-share guarantees are necessary for wireless LANs. In *Proceedings of the SIGOPS European Workshop*, Leuven, Belgium, September 2004.
- [94] Godfrey Tan and John Guttag. Time-based fairness improves performance in multi-rate WLANs. In *Proceedings of USENIX*, Boston, MA, June 2004.

- [95] Godfrey Tan and John Gutttag. The 802.11 MAC protocol leads to inefficient equilibria. In *Proceedings of INFOCOM*, Miami, FL, March 2005.
- [96] Diane Tang and Mary Baker. Analysis of a metropolitan-area wireless network. *Wireless Networks*, 8(2/3):107–120, 2002.
- [97] Y.C. Tay and K.C. Chua. A capacity analysis for the IEEE 802.11 MAC protocol. *ACM/Baltzer Wireless Networks*, 7(2):159–171, Mar 2001.
- [98] Y.C. Tay, Kyle Jamieson, and Hari Balakrishnan. Collision-minimizing CSMA and its applications to wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, August 2004.
- [99] F Tobagi. *Random Access Techniques for Data Transmission over Packet Switched Radio Networks*. PhD dissertation, University of California, Los Angeles, Department of Computer Science, December 1974.
- [100] Fouad Tobagi. Multiaccess protocols in packet communication systems. *IEEE Transactions on Communications*, 28(4), April 1980.
- [101] B. S. Tsybakov and V. A. Mikhailov. Free synchronous packet access in a broadcast channel with feedback. *Prob. Information Transmission*, 14(4):256–280, oct–dec 1978.
- [102] Nitin H. Vaidya, Paramvir Bahl, and Seema Gupta. Distributed fair scheduling in a wireless LAN. In *Proceedings of MOBICOM*, pages 167–178, 2000.
- [103] Wi-Fi Alliance Web Site. <http://www.wi-fi.org>.
- [104] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks, 1995.
- [105] M. Zorzi and R. Rao. Capture and retransmission control in mobile radio. *IEEE Journal on Selected Areas in Communications*, 12(8):1289–1298, 1994.

Appendix A

Supplements to Chapter 3

Our proofs in this appendix is in the context of two nodes, i and j , competing for channel access. We write $G^{1,3}$ to denote the vector of transmission strategy used by i and j such that $g_i = g^1$ and $g_j = g^3$, where g^1 and g^3 transmission strategies. We use \diamond to denote that a wild card strategy. For example, $G^{1,\diamond}$ denotes that $g_i = g^1$ but g_j can be any transmission strategy. We use $\tau(G)$ to denote the duration of a stage game when i and j are competing using strategies, g_i and g_j , specified in G .

A.1 Supplements to Section 3.4

A.1.1 Analysis of DCF

Lemma A.1.4 and **Lemma A.1.5** serve as a more general proof of **Theorem 3.4.3** than the one described **Section 3.4.4**.

Lemma A.1.1. *Under DCF, for any three strategies $g^1 = (r^1, s^1)$, $g^2 = (r^2, s^2)$, and $g^3 = (r^3, s^3)$, where $r^1 > r^2$ and $s^1 = s^2 = s^3 = s$, $f_i(G^{1,3}) < f_i(G^{2,3})$.*

Proof. According to Lemma 3.4.1, $t_i(G^{1,3}) < t_i(G^{2,3})$ and $t_j(G^{1,3}) = t_j(G^{2,3})$. Based on Equation 3.3, we can see that $f_i(G^{1,3}) < f_i(G^{2,3})$.

Lemma A.1.2. *Under DCF, for any three strategies $g^1 = (r^1, s^1)$, $g^2 = (r^2, s^2)$, and $g^3 = (r^3, s^3)$, where $r^1 > r^2$ and $s^1 = s^2 = s^3 = s$, $\gamma^{theo}(g^1) * \frac{t_i(G^{1,3})}{\tau(G^{1,3})} > \gamma^{theo}(g^2) * \frac{f_i(G^{2,3})}{\tau(G^{2,3})}$.*

Proof.

According to Lemma 3.4.1 and Equation 3.3,

$$\gamma^{theo}(g^1) * \frac{t_i(G^{1,3})}{\tau(G^{1,3})} = \frac{s * f^{chan}}{t_i(G^{1,3}) + t_j(G^{1,3})} \text{ and}$$

$$\gamma^{theo}(g^2) * \frac{t_i(G^{2,3})}{\tau(G^{2,3})} = \frac{s * f^{chan}}{t_i(G^{2,3}) + t_j(G^{2,3})}$$

Since $t_j(G^{1,3}) = t_j(G^{2,3})$ and $t_i(G^{1,3}) < t_i(G^{2,3})$ (as evident by Lemma 3.4.1),

$$\gamma^{theo}(g^1) * \frac{t_i(G^{1,3})}{\tau(G^{1,3})} > \gamma^{theo}(g^2) * \frac{f_i(G^{2,3})}{\tau(G^{2,3})}$$

□

Lemma A.1.3. *Under DCF, if node i experiences no losses when transmitting at the highest data rate r^{max} using the maximum frame size s^{max} , the strategy $g^{*i} = (r^{max}, s^{max})$ is the dominant strategy of node i , i.e., $\forall g_i \neq g^{*i}$ and $\forall g_j$, $\gamma_i(g^{*i}, g_j) > \gamma_i(G)$.*

Proof.

$$\begin{aligned} & \forall r_i \neq r^{*i}, r^{*i} > r_i. \text{ Thus, according to Lemma A.1.2,} \\ & \forall g_i \neq g^{*i} \text{ and } \forall g_j, \gamma^{theo}(g^{*i}) * \frac{t_i(G^{*i}, \circ)}{\tau(G^{*i}, \circ)} > \gamma^{theo}(g_i) * \frac{t_i(G)}{\tau(G)} \\ & \text{Since } \alpha_i(g^{*i}) = 1, \\ & \forall g_i \neq g^{*i} \text{ and } \forall g_j \\ & \gamma^{theo}(g^{*i}) * \alpha_i(g^{*i}) * \frac{t_i(G^{*i}, \circ)}{\tau(G^{*i}, \circ)} > \gamma^{theo}(g_i) * \alpha_i(g_i) * \frac{t_i(G)}{\tau(G)} \\ & \text{I.e., } \gamma_i(G^{*i}, \circ) > \gamma_i(G) \end{aligned}$$

□

Lemma A.1.4. *DCF can lead to a unique subgame perfect equilibrium in which a unique NE is played at each stage game.*

Proof. We show that by construction. Assume that both nodes use maximum-sized frames. Also assume that node j has the dominant strategy g^{*j} , i.e., $\forall g_i$ and $\forall g_j \neq g^{*j}$, $\gamma_j(G^{\circ, *j}) > \gamma_j(G)$. $G^{*i, *j}$ forms a unique NE if $\forall g_i \neq g^{*i}$, $\frac{\alpha_i(g^{*i})}{\alpha_i(g_i)} > \frac{\tau(G^{*i, *j})}{\tau(G^{\circ, *j})}$.

Note that the condition $\frac{\alpha_i(g^{*i})}{\alpha_i(g_i)} > \frac{\tau(G^{*i, *j})}{\tau(G^{\circ, *j})}$ is easily satisfied if g^{*i} involves using the highest data rate r^{max} and $\alpha_i(g^{*i}) = 1$. However, in general, that is not the only case. For example, even if $r^{*i} < r^{max}$ and $\alpha_i(g^{*i}) < 1$, the above condition can still hold. Without loss of generality, assume that $r^{max} > r^{*i} > r^{min}$. If node i uses $g^{max} = (r^{max}, s^{max})$, it's possible that $\frac{\alpha_i(g^{*i})}{\alpha_i(g^{max})} > \frac{\tau(G^{*i, *j})}{\tau(G^{max, *j})}$: the right hand side is greater than 1 (since $t_i(G^{\circ, *j}) < t_i(G^{*i, *j})$) but $\alpha_i(g^{*i})$ can be greater than $\alpha_i(g^{max})$.

If i uses $g^{min} = (r^{min}, s^{max})$, it's possible that $\frac{\alpha_i(g^{*i})}{\alpha_i(g^{min})} > \frac{\tau(G^{*i, *j})}{\tau(G^{min, *j})}$. The right hand side is less than 1. As long as $\alpha_i(g_i)$ is not much higher than $\alpha_i(g^{*i})$, g^{*i} can be the dominant strategy. In conclusion, the dominant strategy g^{*i} can constitute any data rate (not just the highest data rate).

□

Lemma A.1.5. *Let there be two possible pairs of strategies (G'^{*j}) and $(G^{*i, *j})$ where $g' \neq g^{*i}$. Furthermore, let $r' > r^{*i}$ and $s' = s^{*i} = s^{*j} = s$. If g^{*i} and g^{*j} are the unique NE strategies under DCF, the NE may be undesirable (and as a result, the subgame perfect equilibrium is also undesirable).*

Informally, this lemma states that if node i and node j use the same frame size and node i is not transmitting at the fastest data rate at equilibrium (i.e., there exists $r' > r^{*i}$), the strategies at the unique NE may be inefficient.

Proof. We prove by showing that for certain combinations of $\alpha_i(g^{*i})$ and $\alpha_i(g')$, it is possible for node i to employ g^{*i} as an equilibrium strategy even though g' yields higher practically achievable throughput, i.e., $\gamma_i^{prac}(g') > \gamma_i^{prac}(g^{*i})$.

Since $(G^{*i, *j})$ are the unique Nash Equilibrium strategies,

$$\gamma_i(G^{*i,*j}) > \gamma_i(G',*j)$$

According to Equation 3.4 and the given assumptions,

$$\gamma_i^{prac}(g^{*i}) * f_i(G^{*i,*j}) > \gamma_i^{prac}(g') * f_i(G',*j)$$

According to Equation 3.5 and Lemma 3.4.1,

$$\begin{aligned} \gamma_i^{theo}(g^{*i}) * \alpha_i(g^{*i}) * \frac{\frac{s}{\gamma_i^{theo}(g^{*i})}}{\tau(G^{*i,*j})} &> \gamma_i^{theo}(g') * \alpha_i(g') * \frac{\frac{s}{\gamma_i^{theo}(g')}}{\tau(G',*j)} \\ \frac{\tau(G',*j)}{\tau(G^{*i,*j})} &> \frac{\alpha_i(g')}{\alpha_i(g^{*i})} \end{aligned}$$

Also, according to Equation 3.1,

$$\begin{aligned} \frac{\tau(G',*j)}{\tau(G^{*i,*j})} &= \frac{f^{chan*} \frac{s}{\gamma_i^{theo}(g')} + \frac{s}{\gamma_i^{theo}(g^{*j})}}{f^{chan*} \frac{s}{\gamma_i^{theo}(g^{*i})} + \frac{s}{\gamma_i^{theo}(g^{*j})}} \\ &= \frac{\frac{1}{\gamma_i^{theo}(g')} + \frac{1}{\gamma_i^{theo}(g^{*j})}}{\frac{1}{\gamma_i^{theo}(g^{*i})} + \frac{1}{\gamma_i^{theo}(g^{*j})}} \end{aligned}$$

Let x be $\frac{1}{\gamma_i^{theo}(g^{*j})}$

$$\frac{\tau(G',*j)}{\tau(G^{*i,*j})} = \frac{1 + \gamma_i^{theo}(g') * x}{1 + \gamma_i^{theo}(g^{*i}) * x} * \frac{\gamma_i^{theo}(g^{*i})}{\gamma_i^{theo}(g')}$$

So, it is possible that

$$\begin{aligned} \frac{\gamma_i^{theo}(g^{*i})}{\gamma_i^{theo}(g')} &< \frac{\alpha_i(g')}{\alpha_i(g^{*i})} \\ \text{since } \frac{1 + \gamma_i^{theo}(g') * x}{1 + \gamma_i^{theo}(g^{*i}) * x} &> 1 \\ \text{If } \frac{\gamma_i^{theo}(g^{*i})}{\gamma_i^{theo}(g')} &< \frac{\alpha_i(g')}{\alpha_i(g^{*i})} \text{ then,} \\ \gamma_i^{prac}(g^{*i}) &< \gamma_i^{prac}(g') \text{ (see Equation 3.5)} \end{aligned}$$

□

Intuitively, node i will use a less efficient data rate r^* as the equilibrium strategy instead of a more efficient strategy r' so long as the proportional increases in the success rate of frame transmission and in the channel times allocated is higher than the proportional reduction in achievable throughput.

A.1.2 Analysis of EDCF

Lemmas A.1.6 and A.1.7 serve as a more general proof of **Theorem 3.4.5** than the one described in in **Section 3.4.5**.

Lemma A.1.6. *EDCF (with FLB or BLB) can lead to a unique subgame perfect equilibrium in which a unique NE is played at each stage game.*

Proof. We prove it by construction. Assume that both nodes use maximum-sized frames. Also assume that node j has the dominant strategy g^{*j} , i.e., $\forall g_i$ and $\forall g_j \neq g^{*j}$, $\gamma_j(G^{*,*j}) > \gamma_j(G)$. $(G^{*i,*j})$ forms a unique NE if $\forall g_i \neq g^{*i}$, $\frac{\alpha_i(g^{*i})}{\alpha_i(g_i)} > \frac{b_i * \tau(G^{*i,*j})}{b^{*i} * \tau(G^{*,*j})}$. According to Lemma 3.4.4 and Equations 3.5, 3.4 and 3.3, it is easy to see that this condition leads to $\gamma_i(G^{*i,*j}) > \gamma_i(G^{*,*j})$.

One example scenario where the necessary condition holds is when $g^{*i} = (r^{max}, s^{max})$, $g^{*j} = (r^{max}, s^{max})$, $\alpha(g^{*i}) = \alpha(g^{*j}) = 1$. Note that this lemma can also hold true in

many scenarios where $r^{*i} \neq r^{max}$ and $\alpha(g^{*i}) < 1$ for reasons similar to those given in Lemma A.1.4. \square

Lemma A.1.7. *Let there be two possible pairs of strategies (G'^{*j}) and $(G^{*i,*j})$ where $g' \neq g^{*i}$. Furthermore, let $r' > r^{*i}$, $s' = s^{*i} = s^{*j} = s$. If g^{*i} and g^{*j} are strategies of a unique NE under EDCF using FLB, the equilibrium may be undesirable. And as a result, the unique SPE is also undesirable.*

Proof. Using a similar procedure described in the proof of Lemma A.1.7, we have

$$\frac{b' + \gamma^{theo}(g') * x}{b^{*i} + \gamma^{theo}(g^{*i}) * x} * \frac{\gamma^{theo}(g^{*i})}{\gamma^{theo}(g')} > \frac{\alpha_i(g')}{\alpha_i(g^{*i})}$$

$$\text{where } x = \frac{b^{*j}}{\gamma^{theo}(g^{*j})}$$

It is possible that $\frac{\gamma^{theo}(g^{*i})}{\gamma^{theo}(g')} < \frac{\alpha_i(g')}{\alpha_i(g^{*i})}$ since

$$\frac{b' + \gamma^{theo}(g') * x}{b^{*i} + \gamma^{theo}(g^{*i}) * x} \text{ can be greater 1.}$$

Intuitively, if node i transmits at high data rate r' and the loss rate experienced is high, node i will not be able to transmit the maximum number of frames allowed under t^{max} . Therefore, if node i , by transmitting at a lower data rate r^{*i} , can reduce the loss rate low enough such that b^{*i} is larger than b' , the node will prefer to use r^{*i} over r' even though $\gamma_i^{prac}(g') > \gamma_i^{prac}(g^{*i})$. \square

Appendix B

Supplements to Chapter 5

B.1 Discussion on Analysis and Validation

In this section, we revisit the assumptions surrounding Equation 5.2 and discusses the accuracy of our analysis. We then validate our average-case analysis through simulation.

We restate Equation 5.2:

$$Pcol_B = \frac{1}{CWa_A + 1}$$

which describes the probability of node B 's transmission colliding with that of node A . This equation is correct when node B has an equal probability to transmit at each timeslot in the idle-busy timeline. However, this assumption may not be entirely accurate when the backoff counter is drawn from a uniform distribution of $Base$ and CW , where $Base = 0$ and CW is small. This is because, when node A picks 0, it will transmit a frame immediately whereas node B 's backoff counter remains unchanged. Therefore, the assumption that node B will transmit with an equal probability at the beginning of each time slot is not entirely accurately since the probability of node B transmitting at the beginning the 0^{th} slot could be different from that of other slots. However, this affect becomes insignificant with sufficient large CW , which is the case in practice. This situation does not exist when the backoff counter is drawn randomly from a uniform distribution of $Base = 1$ and CW . When $Base = 1$, a node can only transmit at the end of each idle slot and thus $Pcol_B = \frac{1}{CWa_A}$. Therefore, the general equation that applies when $Base = 0$ or $Base = 1$ is:

$$Pcol_B = \frac{1}{CWa_A + 1 - Base} \quad (B.1)$$

For any contender i ,

$$CWa_i = \frac{CW_i + Base_i}{2} \quad (B.2)$$

where $Base_i = 0$ or $Base_i = 1$. Our analysis in Section 5.2 applies in both cases.

B.1.1 Validation

We write a simple program to simulate multiple node competing for channel access. Each node attempts to transmit about 5000 frames. We examine the max/min ratio of the analytically derived value and the value obtained through simulation for each the four different variables: i) $Pcol_{all}$ or $Pcol$, the ratio of the number of collided frame transmissions to the total number of frame transmissions, ii) $Ntxpercol$, the average number of frame transmissions involved in each collision event, iii) $Ftxev$, the ratio of the number of transmission events to the number of frame transmissions, and iv) $Tidle$, the average amount of idle time per transmission events.

As shown in Figure B-1(a), when there are 2 competing nodes using the same value of CW and $Base = 1$, the max/min ratios are virtually 1. When there are 4 contenders, the max/min ratio of $Pcol_{all}$ is roughly 1 whereas the ratios of the other three variables are greater than 1 at low CW values (see B-1(b)). This is because $Ntxpercol > 2$ when the collision rate is high (as a result of small CW values). Therefore, the assumption of $Ntxpercol = 2$ used in our equations is inaccurate. As a result, the analytically derived values of $Ftxev$ and $Tidle$ are relatively far from the simulation results. However, in practice, our desired collision rate Figures B-2(a) and B-2(b), plot the absolute values of $Pcol_{all}$ and $Ntxpercol$ obtained through simulation. As shown in these figures, the assumption that $Ntxpercol \sim 2$ is fairly accurate whenever $Pcol_{all} \leq 0.25$, which is the case in practice.

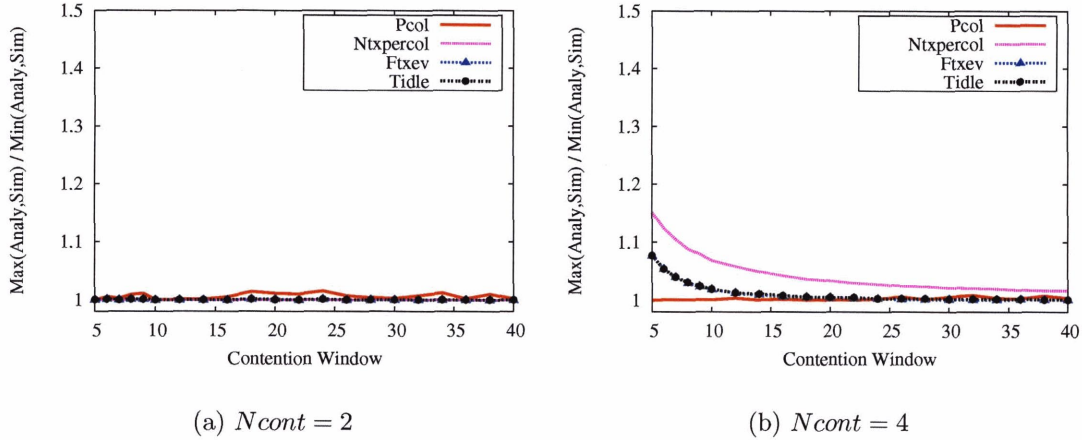


Figure B-1: Ratios of the value obtained through simulation and the analytically derived value of $Pcol_{all}$, $Ntxpercol$, $Ftxev$ and $Tidle$, when 2 or 4 continuously backlogged nodes compete for channel access using the same CW and $Base = 1$.

Figures B-3(a) and B-3(b) plot the max/min ratios with $Ncont = 2$ and $Ncont = 4$ when $Base = 0$ is used. In both figures, the max/min ratios of $Ntxpercol$, $Ftxev$ and $Tidle$ follow those in Figures B-1(a) and B-1(b). The only exception is that $Pcol_{all}$ is relatively higher than 1, especially at smaller CW s for the reasons explained earlier about issues surrounding the usage of $Base = 0$. However, notice that the max/min ratios of $Ftxev$ and $Tidle$ are very close to 1.

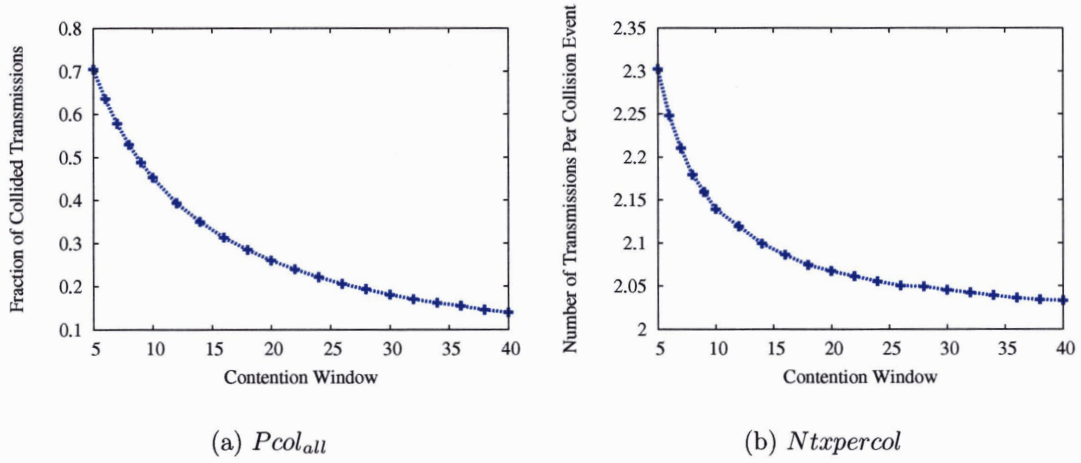


Figure B-2: Absolute values of $P_{col_{all}}$ and $N_{txpercol}$ obtained throughput simulation when 4 continuously backlogged nodes compete for channel access using the same CW and $Base = 1$.

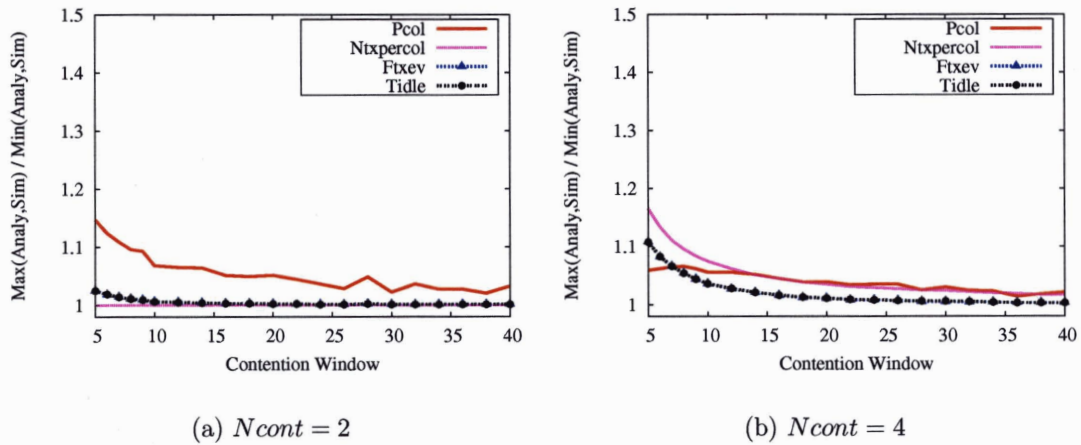


Figure B-3: Ratios of the value obtained through simulation and the analytically derived value of $P_{col_{all}}$, $N_{txpercol}$, F_{txev} and T_{idle} , when 2 or 4 continuously backlogged nodes compete for channel access using the same CW and $Base = 0$.